
FEBRERO 1988
REVISTA PARA USUARIOS DE LA NORMA
LOAD MSX
AÑO 2 Nº 22
A 8 REP.ARGENTINA
ESTUDIANDO EN VERANO
CONCURSO Y SORTEO
INEDITOS
•COPIADOR DE PANTALLAS
• COMO HACER UNA BASE DE DATOS
• 4 EN UNA LINEA

A la computadora personal

Talent MSX

nada le es imposible

dialogo - D. P.



MSX es marca registrada de MICROSOFT CORPORATION

Porque gracias a la norma internacional MSX, la TALENT MSX trasciende sus propios límites.
Hasta ahora, cuando usted compraba una computadora personal de cualquier marca, quedaba automáticamente desconectado del resto del mundo de la computación. Porque los distintos equipos y sistemas no eran compatibles entre sí.
Hasta que dos grandes empresas de informática, la Microsoft Corp. de EE.UU. y la ASCII del Japón se pusieron de acuerdo para crear una norma standard: la MSX. Que se expandió también rápidamente en Europa. Y que hoy TALENT presenta por primera vez en la Argentina.
Mientras que la mayoría de las computadoras de su tipo que se ofrecen en el mercado nacional, han sido discontinuadas por obsoletas en sus lugares de origen, TALENT MSX tiene casi ilimitadas posibilidades de desarrollo. Porque la norma MSX es en todo el mundo inteligencia en crecimiento.
La TALENT MSX pone a su disposición un mundo de software para elegir. Y con la incorporación de todos sus periféricos llega a ser una auténtica computadora profesional.

UTIL

Su poderoso sistema operativo MSX permite el acceso a todo tipo de procesamiento de datos:

- Planillas de calculo.
- Procesadores de palabra.
- Graficos de negocios.
- Bases de datos (d Base II, etc.)
- Contabilidad general, sueldos, y jornales, costos, etc., desarrollados bajo CP/M en Basic, Cobol, Pascal o C.

Con la posibilidad de conexion a linea telefonica permite la transferencia y consulta de datos entre computadoras personales, profesionales o bancos de datos.
La grabacion de archivos es en formato MS-DOS, haciendola compatible con las computadoras profesionales.

DIDACTICA

Dispone de tres lenguajes para la enseñanza de computacion: LOGO como lenguaje de induccion para los mas chicos. Lenguaje de Programacion en castellano, para todos los que quieran aprender a programar sin conocimientos previos. Y Basic MSX como lenguaje profesional. Mas una amplia variedad de perifericos como el Mouse, Lapiz Optico, Tableta grafica, Track-ball, etc.

DIVERTIDA

La mas genial para Video-Juegos. Por la amplisima biblioteca de programas **–todos nuevos–** de la norma MSX en el mundo. Y ademas, el Basic MSX permite al usuario generar sus propios juegos con un manejo tan simple, como sólo TALENT MSX puede ofrecer.

CARACTERISTICAS TECNICAS

- Memoria principal 64 KB ampliable hasta 576 KB.
- Memoria de video: 16 KB RAM.
- ROM incorporada de 32 KB con el MSX-Basic de Microsoft.
- Graficos completos, hasta 32 sprites y 16 colores simultaneos.
- Generador de sonido de 3 voces y 8 octavas.
- Conexion para cualquier grabador.
- Interfaz para salida impresora paralela.
- Conectores para cartuchos y expansiones.
- Fuente para 220 V y modulador PAL-N incorporado.

DISTRIBUIDORES AUTORIZADOS: CAPITAL FEDERAL: AMATRIX, Bolívar 173 - ARGECINT, Av. de Mayo 1402 - BAIDAT COMPUTACION, Juramento 2349 - COMPUPRANDO, Av. de Mayo 965 - COMPUSHOP, Córdoba 1464 - COMPUTIQUE, Córdoba 1111, E. P. - COMPUTRONIC, Viamonte 2096 - CP67 CLUB, Florida 683, L. 18 - DALTON COMPUTACION, Cabildo 2283 - ELAB, Cabildo 730 - MICROSTAR, Callao 462 - Q.S.P., Bartolomé Mitre 864 - SERVICIOS EN INFORMATICA, Paraná 164 - DISTRIBUIDORA CONCALES, Tucumán 1458 - MICROMATICA, Av. Pueyrredón 1135 - **ACASSUSO:** MICROSTAR ACASSUSO, Eduardo Costa 892 - **AVELLANEDA:** ARGOS, Av. Mitre 1755 - **BOULOGNE:** COMPUTIQUE CARREFOUR, Bernardo de Irigoyen 2647 - **CASTELAR:** HOT BIT COMPUTACION, Carlos Casares 997 - **LANUS:** COMPUTACION LANUS, Caaguazú 2186 - **LOMAS DE ZAMORA:** ARGESIS COMPUTACION, Av. Meeks 269 - **MARTINEZ:** VIDEO BYTE, Hipólito Yrigoyen 32 - **RAMOS MEJIA:** MANIAC COMPUTACION, Rivadavia 13734 - **SAN ISIDRO:** FERNANDO CORATELLA, Cosme Beccar 249 - **VICENTE LOPEZ:** SERVICIOS EN INFORMATICA, Av. del Libertador 882 - **BAHIA BLANCA:** SERCOM, Donado 327 - SUMASUR, Alsina 236 - **LA PLATA:** CADEMA, Calle 7 N° 1240 - CERO-UNO INFORMATICA, Calle 48 N° 529 - **MAR DEL PLATA:** FAST, Catamarca 1755 - **NECOCHEA:** CAFAL, Calle 57 N° 2920 - SERCOM, Calle 57 N° 2216 - **TRENQUE LAUQUEN:** COMPUQUEN, Villegas 231 - **CORDOBA:** AUTODATA, Pasaje Santa Catalina 27 - TECSIEM, Santa Rosa 715 - **ROSARIO:** 2001 COMPUTACION, Santa Fe 1468 - MINICOMP, Maipú 862 - SISOR, Urquiza 1062 - **SANTA FE:** ARGECINT, P. San Martín 2433, L. 36 -SISOR, Rivadavia 2553- INFORMATICA, San Gerónimo 2721/25 - **VILLA MARIA:** JUAN CARLOS TRENTO, 9 de Julio 80 - **MENDOZA:** INTERFACE, Sarmiento 98 - BIT & BYTE, 9 de Julio 1030 - **COMODORO RIVADAVIA:** COMPUSER, 25 de Mayo 827 - **GENERAL ROCA:** DISTRIBUIDORA VECCHI, 25 de Mayo 762 - **LA PAMPA:** MARINELLI, Pellegrini 155 - **NEUQUEN:** MEGA, Perito Moreno 383 - EDISA, Roca esq. Fotheringham - **RIO GRANDE:** INFORMATICA M & B, Perito Moreno 290 - **SAN CARLOS DE BARILOCHE:** L. ROBLEDO & ASOCIADOS, Elfein 13, Piso 1° - **TRELEW:** SISTENOVA, Sarmiento 456 - **PARANA:** MARIO GARCIA, Laprida y Santa Fe - **POSADAS:** CENTRO DE COMPUTOS ELDORADO, Colón 2429 - **RESISTENCIA:** FRANCO SANTI, Carlos Pellegrini 761 - **SAN SALVADOR DE JUJUY:** DELTA COMPUTACION, Salta 1108 - **SALTA:** DELTA COMPUTACION, Caseros 873 - **SAN MIGUEL DE TUCUMAN:** LEXICON, 9 de Julio 85 - ELECTRONICA

STAFF

LOAD MSX

Director Periodístico

Fernando Flores

Secretario de Redacción

Ariel Testori

Redacción

Fernando Pedró

Arte y Diagramación

Fernando Amengual y Tamara Migelson

Departamento de Avisos

Oscar Devoto y Nelzo Capello

Departamento de Publicidad

Dolores Urien

Servicios Fotográficos

Image Bank, Oscar Burriel y Victor Grubicy

EDITORIAL PROEDI

Presidente

Ernesto del Castillo

Vicepresidente

Cristian Pusso

Director Titular

Javier Campos Malbrán

Director Suplente

Armengol Torres Sabaté

Load Revista para usuarios de la norma MSX es una publicación mensual editada por Editorial PROEDI S.A., Paraná 720, 5° Piso, (1017) Buenos Aires. Tel.: 46-2886 y 49-7130. Radiollamada: Tel.: 311-0056 y 312-6383, código 5941. Registro Nacional de la Propiedad Intelectual: E. T. M. Registrada. Queda hecho el depósito que indica la Ley 11.723 de la Propiedad Intelectual.

ISSN 0326-8241

Impreso en Impresiones Gráficas Tabaré S.A.I.C. Erézcano 3158 Cap. - Fotocromo tapa: Columbia. Fotocomposión: Interamericana Gráfica.

Distribuidor en Capital: Martino, Juan de Garay 358, P. B. Capital.
Distribuidor interior: D G P: Hipólito Yrigoyen 1450, Capital Federal. T.E. 38-9266/9800.

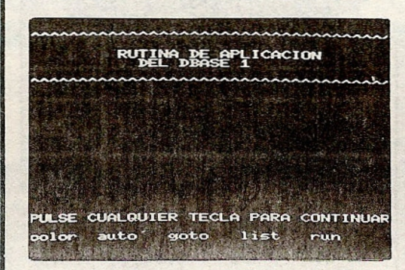
CORREO ARGENTINO SUC. CASA CENTRAL | FRANQUEO PAGO CONCESION N° 2550 | TARIFA REDUCIDA CONCESION N° 1304

SUMARIO

UN BOY SCOUT HECHO DE BITS

La concepción de la computación hogareña como causante de la pérdida de tiempo por tareas recreativas muchas veces confunde a los potenciales usuarios. Esa idea no permite verla como una verdadera ayuda para el estudiante secundario, como lo demuestra este artículo. **(Pág. 6)**

DBASE I



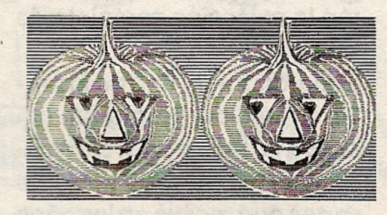
Si bien es común ver publicados programas de archivos simples para los usos más diversos, es notoria la falta de un sistema que sea realmente eficiente. Este programa, ganador del segundo premio del 3er. Concurso de Programas, trata de cubrir ese campo. **(Pág. 8)**

LOS CICLOS WHILE... DO...

La realización de una tarea cualquiera por medio de un sistema informatizado incluye una serie de pasos, entre los cuales la ejecución del programa codificado en BASIC no es sino el último de ellos. **(Pág. 14)**

CUATRO EN LINEA

Un apasionante juego de estrategia para dos jugadores, con buenos efectos visuales y sonoros. Mención del 3er. Concurso de Programas. **(Pág. 18)**

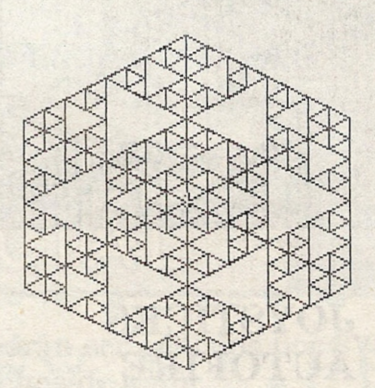


ASSEMBLER, BYTES Y OTRAS YERBAS

(3ra. parte)

Continuamos con la serie de artículos para poder programar en este potente lenguaje. **(Pág. 20)**

RINCON DEL USUARIO



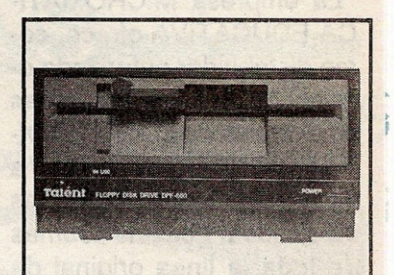
Lucas Film Inc., el estudio cinematográfico que realiza los efectos especiales de las películas de George Lucas, utiliza para ello una de las más modernas computadoras. Una de las técnicas para generar gráficos es con fractales. De esta forma es posible la generación de montañas, lagos, nubes, y otros muchos elementos de la escenografía. **(Pág. 22)**

PROGRAMAS

dBASE I (Pág. 80) - Cuatro en línea (Pág. 18) - Hard Copy (Pág. 30)

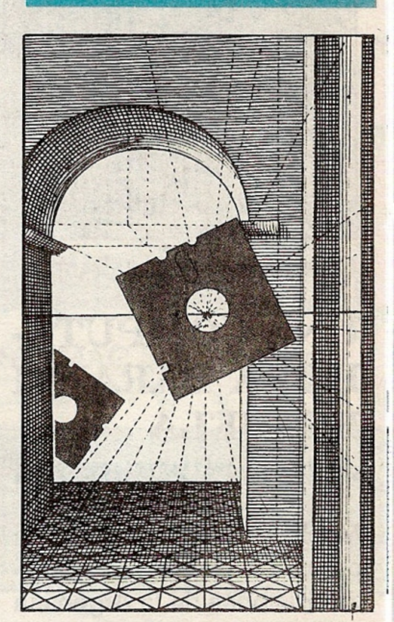
LOS MISTERIOS DEL MSX-DOS: OPERACION

(1ra. parte)



En el número 20 hicimos la presentación, ahora nos dedicaremos al análisis del otro lado del sistema. Esta vez desde el punto de vista del usuario avanzado, para que quede claro el entorno de trabajo que brinda la disquetera **(Pág. 26)**

HARD COPY



Este programa se encarga de copiar una pantalla gráfica al papel con una impresora matricial. **(Pág. 30)**

SECCIONES FIJAS

Noticias MSX (Pág. 4) - Soft al día (Pág. 32) - Buzón (Pág. 34)

LOGO TRIDIMENSIONAL

La empresa MICROMATICA EDUCATIVA ofrece, como distribuidor autorizado oficial TALENT, los programas adicionales de LOGO TRIDIMENSIONAL para aplicar a dicho paquete, en versiones de disquete y casete además de toda la línea original de software para la norma MSX así como consolas y periféricos de la nueva MSX II.



Eduardo Vaca y Otelo Borroni.

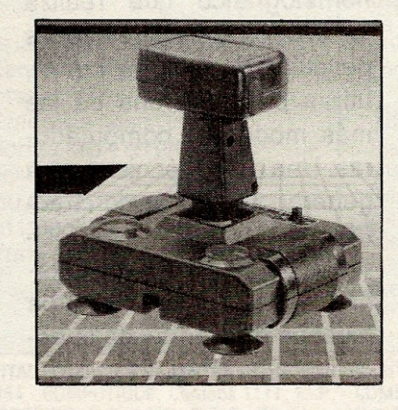
UNA COMPUTADORA PARA MI ESCUELA

Muchas cartas se recibieron para el certamen que organiza "Historias de la Argentina secreta", con el auspicio de la revista K64, que permitirá que dos escuelas argentinas posean un equipo completo de computación Talent MSX y suscripciones de la revista K 64. Los alumnos deben hacer llegar una carta -antes de fin de marzo- a "Historias de la Argentina secreta", ATC, Avda. Pte. Figueroa Alcorta 2977, (1425) Buenos Aires. En la misma deberán indicar nombre y apellido, nombre de la escuela a la que concurren, grado y dirección del establecimiento.

Además, las primeras cien escuelas que escriban recibirán la colección completa de K64.

Eduardo Vaca, uno de los realizadores del programa, utiliza ya la Talent para almacenar los datos de los miles de cartas que recibió en los años que lleva el prestigioso programa, y para contestar la correspondencia. Además hay importantes proyectos que interconectarán a la computadora con el programa. Pero eso, por ahora, es una sorpresa.

JOYSTICKS AUTOFIRE

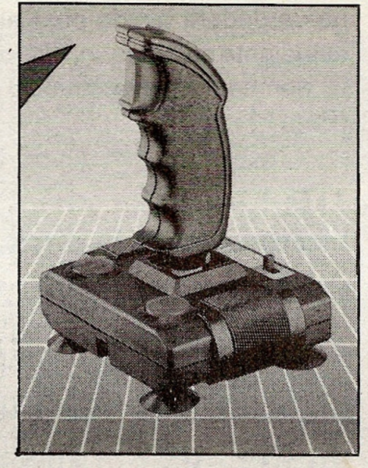


En materia de joysticks nunca estará dicha la última palabra.

Los gustos cambian, y los joysticks también. Estos dos modelos se pueden obtener para MSX.

Las principales características de estos joysticks son las siguientes:

- Cuatro disparadores.
- Control del movimento en ocho direcciones distintas, con retorno automático al centro si se lo deja libre.
- Puede ser utilizado tanto por personas diestras como zurdas, gracias a la disposición de los disparadores.



- Interruptor que permite conectar o desconectar el disparador automático, muy práctico para no tener que estar presionando continuamente el botón de disparo.

La construcción de los joysticks es muy sólida, y éstos dan la impresión de ser bastante "duros".

En las pruebas realizadas en nuestra editorial,los mismos demostraron soportar toda clase de abusos, sin el menor síntoma de daño o mal funcionamiento.

MSX PROGRAMAS Y UTILIDADES

por: Luers

En este libro encontraremos una sucesión de programas que apoyan las explicaciones de conceptos cuyo conocimiento es fundamental para los futuros programadores.

Los programas no se explicarán línea por línea. En el texto se apuntan detalles interesantes, además de haber sentencias REM explicativas en los listados.



Otro punto importante es que ayudará al usuario de estas máquinas a tener una idea de cómo funcionan "interiormente".

Para aprovechar más las posibilidades de nuestra computadora, fueron desarrollados editores de gráficos y sonidos.

Otros programas que servirán para enseñar a los programadores sin mucha experiencia a conocer mejor sus computadoras son el diseñador de caracteres, gestión de datos en cinta o disco, diagramador de barras, descifrador de códigos y convertidor de valores a diferentes sistemas numéricos.

Uno de los programas más valiosos incorporados en esta obra es un desensamblador que permitirá al lector dar los primeros pasos en la programación en código de máquina.

Otro software de gran utilidad para el programador en BASIC es el de "manejo de errores". Este amplía la información del error que se cometió en la construcción del programa.

Al comienzo del libro, encontraremos las diferencias entre las Spectravideo 318/328 y las demás MSX para conseguir adaptar los programas de una máquina a otra.

Junto con los programas se dan consejos y trucos importantes para tener presentes al crear nuestros propios juegos o utilitarios.(Edita: Ferre Moret S.A.. Distribuye: Data Becker)

4to CONCURSO

DE PROGRAMAS

auspiciado por TELEMATICA S.A. que proveerá los siguientes Premios:

PRIMER PREMIO

UN PERIFERICO

(a elección entre un monitor, y una disquetera y una impresora).

SEGUNDO PREMIO

UN PERIFERICO

(a elección entre un monitor, una disquetera y una impresora).

ESPECIAL

Entre los programas recibidos, algunos de ellos podrán ser editados por SYSTEMAC S.A., reconociéndose los derechos de autor.

Se premiará el mejor software de cualquier clase (juegos, utilitarios, científico o comercial).

BASES

No sólo será indispensable que el programa enviado en casete ó disquete funcione correctamente, sino que además debe cumplir con ciertas reglas:

.Programación estructurada en bloques fácilmente diferenciables.
.Fácil seguimiento del mismo y detalle de éste como parte de su documentación. (Diagrama de bloques con los números de línea que los identifiquen).
.Aclaración y clara explicación de los algoritmos utilizados, deben figurar como parte de la documentación.
.Las variables y/o direcciones de memoria utilizados también se deben incluir en esta documentación.
.Listado de nemónicos assembler y la localización en memoria si es que se utiliza este tipo de lenguaje.
.Calidad y originalidad de gráficos, sonidos y pantallas de menú.

Los trabajos deberán enviarse antes del 30 de julio próximo (cierre del certámen) a: Paraná 720, piso 5, (1017) Capital Federal.

UN BOY SCOUT HECHO DE BITS

La concepción de la computación hogareña como causante de la pérdida de tiempo por tareas recreativas muchas veces confunde a los potenciales usuarios. Esa idea no permite verla como una verdadera ayuda para el estudiante secundario, como lo demuestra este artículo.

Para aquellos lectores que se encuentren cursando el colegio, ya sea primario o secundario, se acerca una época no muy dichosa como es la de exámenes pendientes del año pasado . Contrario a ciertas costumbres habituales, no debe descartarse completamente el uso de la computadora hogareña con vistas a proveer un cierto apoyo que puede hacer más fluída la tediosa tarea de la preparación de algunas materias, si se sabe administrar la aplicación del mismo en rubros de por si engorrosos . La famosa frase de "¡¡¡Hasta que no apruebes el examen , nada de computadora !!! " no siempre es una contribución para obtener mejores resultados con el adolescente.

Aunque parezca mentira , suele perderse mucho tiempo en labores como los largos cálculos, que son a veces necesarios de efectuar a mano o mediante calculadoras no programables, en materias que exigen la resolución numérica como único comprobante de haber planteado correctamente el problema.

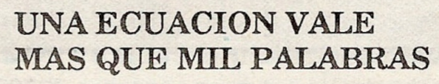
Es en estos casos donde se puede lograr una mayor dedicación al estudio de cada variante posible , como camino para llegar al resultado final , programando la/s formula/s necesaria/s por algún medio.

De modo que solamente reste introducir valores nuevos o con alguna variante para obtener a " toque de tecla " su cálculo .

Aquí es donde aparece la ventaja obvia de una computadora : es programable , hace cálculos numéricos de lo más engorrosos (antes de ejercer una sonrisa socarrona piense si a Usted se le había ocurrido) , y como si todo esto fuera poco , habita en el seno de nuestro hogar .

Naturalmente , siempre hay una restricción en todo maravilloso descubrimiento; en este caso depende de la fuerza de voluntad de cada uno y de la honestidad para no sucumbir ante la tentación de colocar algun casete o disco de juegos en lugar de las odiosas ecuaciones .

UNA ECUACION VALE MAS QUE MIL PALABRAS



En efecto , no sería demasiado clarificador extenderse "filosóficamente " sobre el tema sin un ejemplo . Vayamos al mismo:

Supongamos dada la siguiente ecuación : 10+X-2=3+2x5 , hallar el valor de X.

1.- Despejar X : X=(3+2x5)+2-10

2.- Calcular X : X=13+2-10=5

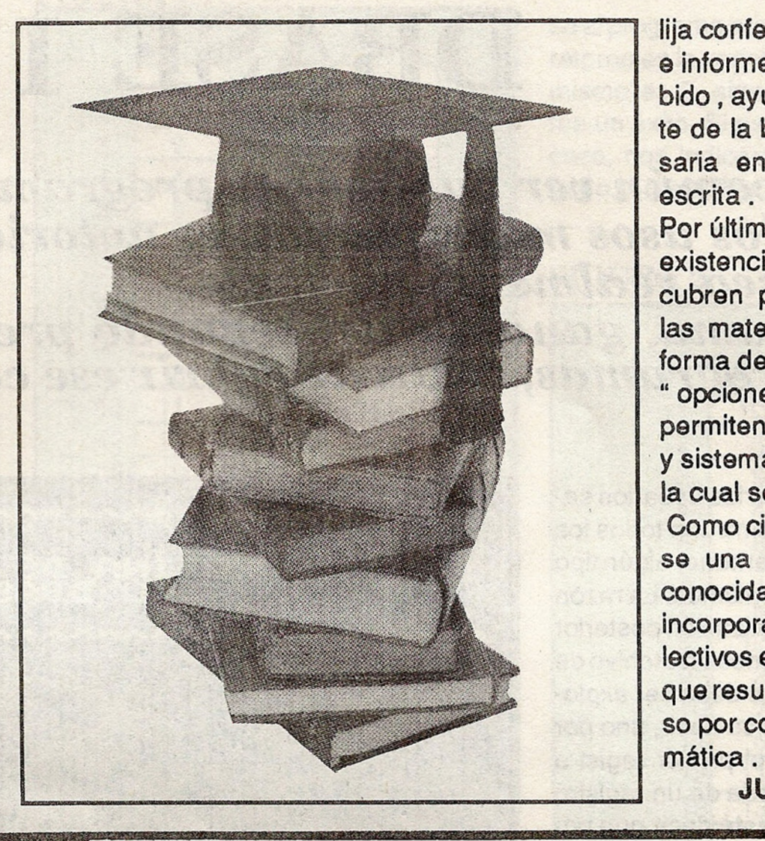
3.- Verificar : 10+5-2=3+2x5 => 13=13

Si Usted es de los que calculó el valor X=17 ; debo decepcionarlo ya que cometió un error muy habitual . El término de la derecha , entre paréntesis en el paso 1 , debe calcularse como 3+(2x5) y no como (3+2)x5 en virtud de la jerarquía de las operaciones que seguramente conoce pero no siempre aplica (¡grave error !) . El paso 3 también podría haberse efectuado escribiendo la igualdad tal cual se encuentra precedida de una instrucción PRINT ; obteniéndose como respuesta

por parte de la computadora , luego de presionar <ENTER> , el número -1 . No piense que su computadora delira sino todo lo contrario , este número es utilizado para denotar una condición verdadera (TRUE) en respuesta a la igualdad planteada y por lo tanto indicar que es correcta .

Las aplicaciones matemáticas de este tipo se extienden a una importante porción de materias como lo son todas aquellas que involucren cálculos de fórmulas , desde la matemática más elemental a la física y las materias de raíz contable . No debe tampoco olvidarse la posibilidad de graficar en pantalla , mediante simples programas de pocas líneas , curvas desconocidas y así conocer sus particularidades .

Los que deban emitir informes y además sean poseedores de una impresora pueden encontrar en los graficadores y procesadores de texto una herramienta importante y muy poderosa para una prolija confección de resúmenes e informes que , como es sabido , ayudarán a brindar parte de la buena imagen necesaria en toda presentacion escrita .



Por último cabe mencionar la existencia de programas que cubren prácticamente todas las materias , realizados en forma de cuestionarios tipo " opciones múltiples " , y que permiten un repaso ordenado y sistemático de la ciencia de la cual se ocupan .

Como cierre no debe olvidarse una materia sumamente conocida , cada vez de mayor incorporación en los ciclos lectivos en forma obligatoria y que resulta ideal para su repaso por computadora : la informática .

JUAN PABLO BAUER

DBASE I

Si bien es común ver publicados programas de archivos simples para los usos más diversos, es notoria la falta de un sistema que sea realmente eficiente.
Este programa, ganador del segundo premio del tercer concurso de programas, trata de cubrir ese campo.

Como casi todos los usuarios sabrán, el problema con todos los archivos es mantener algún tipo de orden dentro de los mismos. La razón de esto radica en la localización posterior de un registro. Imaginemos un archivo de unos 1000 registros que deba ser explorado secuencialmente, es decir, uno por uno comenzando por el primer registro del archivo, en búsqueda de un registro en particular. De más está decir que podemos ir a dar una vuelta mientras la máquina realiza el trabajo.

La otra posibilidad es que el archivo sea ordenado cada vez que se le realiza una modificación. Este procedimiento se denomina "SORT". En este caso, cuando el archivo crece lo suficiente, no sólo podemos ir a dar una vuelta, sino que podemos dar varias.

El método utilizado a continuación es conocido como acceso por clave, o índice, o "indexación" de un archivo.

La idea consiste en utilizar un campo como clave de acceso al registro.

Para quienes no estén al tanto de estos téminos, podemos hacer una simple analogía con lo que ocurre en el fichero de una biblioteca.

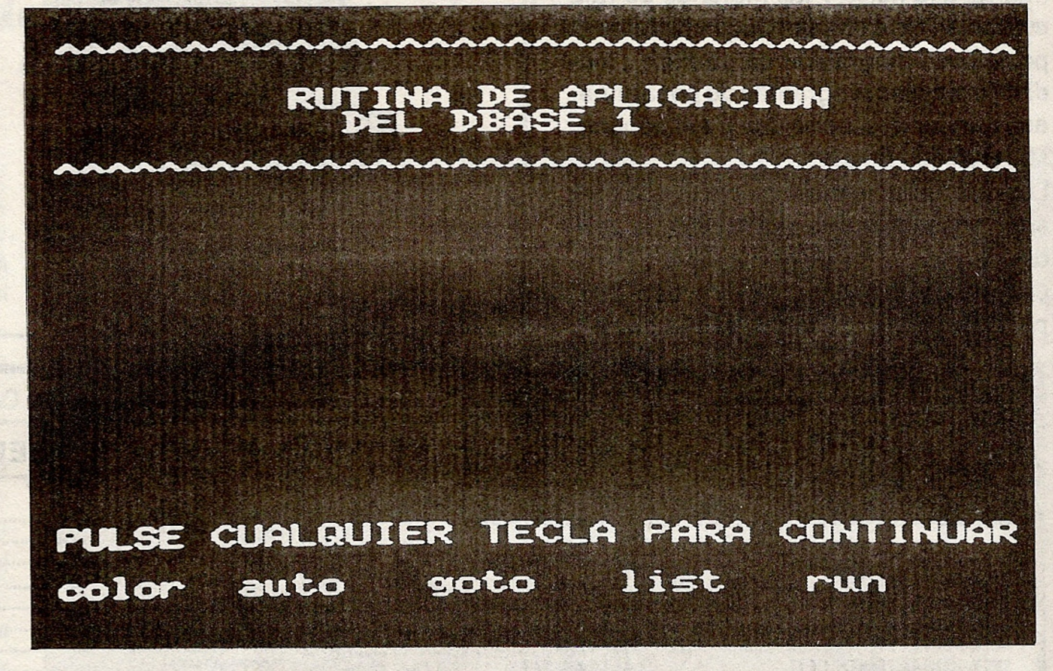
Lo que llamamos registros es cada una de las fichas; mientras que los campos son los diferentes items que contiene cada ficha.

Como el archivo índice reside en la memoria, la búsqueda no demora más de 2 segundos, aun con archivos grandes.

Luego, con el valor correspondiente a la clave, accedemos al archivo de datos y asunto arreglado.

Las principales características de este programa (que en realidad es una gran subrutina) son:

-UTILIZA CONCEPTOS DE PROGRAMACION ESTRUCTURADA.
-INFINITAS POSIBILIDADES DE APLICACION.
-TIENE EL CORAZON DEL DBASE II
-COMO EN DBASE, NO ES NECESARIO QUE EL USUARIO ENTIENDA QUE SUCEDE DENTRO DE LAS RUTINAS.
-ES DE 10 A 50 VECES MAS RAPIDO QUE OTROS METODOS.
-APROVECHA AL MAXIMO LAS CARACTERISTICAS DEL EQUIPO.
-PERMITE CREAR CON MUY POCO ESFUERZO.
-UTILIZA VARIABLES LOCALES, PARA AHORRAR MEMORIA Y EVITAR CONFUSIONES CON LAS VARIABLES DEL PROGRAMA PRINCIPAL.



EXPLICACION DEL FUNCIONAMIENTO DE LAS RUTINAS.

La estructura de este sistema de archivos está formada por un archivo que contendrá los datos, y otro que actuará como índice para el primero.

En el caso del ejemplo que les envío, se trata de un sistema de archivos que mantiene al día a los suscriptores de una revista, que bien podría ser LOAD MSX.

El archivo de dato es un archivo simple, de acceso aleatorio, que puede ser definido por el usuario sin mayores complicaciones. Normalmente, los registros son almacenados físicamente en el mismo orden en que son ingresados al archivo. La única excepción se produce cuando se utiliza algún espacio que fue dejado vacante por un registro borrado.

El archivo índice contiene una serie de punteros de los datos, que permiten acceder a los mismos en orden ascendente o descendente. El programador sólo le debe dar el nombre a este archivo, ya que su estructura es determinada por el programa.

Internamente, se almacenan 64 punteros en cada registro del archivo índice. Una vez que terminan los punteros en el archivo índice, se guardan otros tres datos.

El último campo de todo el archivo índice contiene la cantidad de registros activos del archivo (datos). El anterior nos dice la cantidad de registros borrados del archivo, y el anterior a éste será un puntero tipo stack LIFO (Last In First Out) hacia el último registro borrado.

La correlación entre el archivo de datos y el índice se realiza por medio de dos vectores KE$ y PT%. Estos son utilizados

por las rutinas para mantener el orden del archivo, y de esta forma poder acceder rápidamente a un registro.

Para comprender mejor cómo funciona este sistema, podemos ver la figura 1. Mientras el vector KE$ contiene el campo clave sin ningún tipo de orden (tal como está en el archivo), el vector PT% contiene una lista de punteros que señalan los valores de KE$ en orden ascendente. El vector PT% contendrá exactamente la misma información que el archivo índice, incluidos los últimos valores de registros activos y borrados. Para entender la lógica de esta estructura de datos, veamos qué sucede cuando un registro es agregado al archivo.

La figura 2 nos muestra cómo queda el vector PT% despues de agregar un registro cuyo valor de campo clave fue 50. Notemos los siguientes cambios:

-El séptimo lugar del vector KE$ que antes tenía una clave ya borrada contiene ahora la nueva clave correspondiente al nuevo registro.

-El número total de registros borrados (PT% (255)) fue disminuido en I.

-El total de registros (PT% (256)) fue incrementado en I.

-Un puntero que indicaba la posición del nuevo registro fue insertado en la dirección I2 de PT%.

Para borrar un registro, el proceso es invertido.

Es importante notar que todos los cambios que se realizan sobre el vector PT% no son llevados inmediatamente al archivo índice. Todas las modificaciones ocurren dentro de la memoria de la máquina. Como consecuencia de esto, todos estos cambios deben ser almacenados antes de finalizar las operaciones.

Lógicamente, todas estas operaciones son "transparentes" para el usuario, ya que él no se entera de nada.

Finalmente, para localizar un registro por su clave, se realiza una búsqueda dentro del vector PT%.

Primero nos situamos en el medio del mismo, y verificamos si la clave encontrada coincide con la que buscamos. Si esto no sucede, volvemos a partir al medio el vector PT%, y comparamos nuevamente. Este proceso se repite hasta que se encuent re una coincidencia.

Como las claves son buscadas dentro de la memoria de la máquina, no es necesario acceder al disco, realizando de este modo un ahorro considerable de tiempo.

Sólo cuando encontramos la clave buscada vamos a acceder al disco, y la posición del registro dentro del archivo estará indicada por el valor de PT%.

FIGURA 1

	PT%()	KE$()
1	10	022
2	3	049
3	9	003
4	4	010
5	8	034
6	1	071
7	5	098
8	12	015
9	2	004
10	6	001
11	11	084
12	13	046
13	0	099
	0	

7	Puntero al Reg. borrado
1	Nro. de Reg. borrados
12	Nro. de Reg. Activos

El sistema de stack para los registros borrados nos permite eliminar el desperdicio de lugar en el archivo de datos, ya que cada vez que borramos un registro, éste no es eliminado físicamente del mismo, y por lo tanto el lugar que ocupa se desperdicia.

Ahora que ya sabemos qué sucede dentro de las rutinas, veamos cómo utilizar las mismas.

UTILIZANDO LAS RUTINAS DEL DBASE I

Como dijimos en un principio, es bueno saber todo lo explicado, pero se vuelve totalmente innecesario a la hora de utilizar el programa. DBASE I nos permite realizar 8 operaciones distintas con nuestro archivo de datos. Estas operaciones están disponibles para cualquier programa escrito en BASIC, ya que aquí no hay llamadas extrañas a rutinas en código máquina, ni cosas por el estilo.

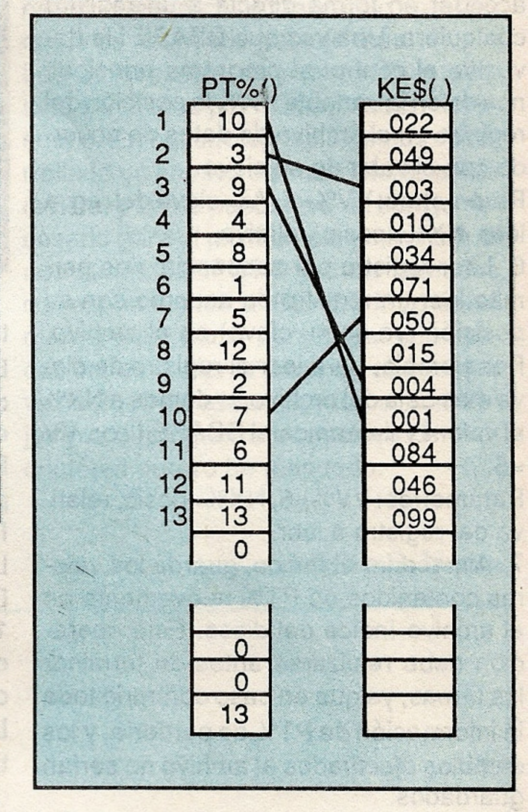
En la tabla I tenemos un resumen de las 8 operaciones, y qué variables utilizar.

Para usar el DBASE I, debemos darle a la variable VV% el valor correspondiente a la operación deseada, y luego llamar a la rutina mediante un GOSUB.

Una vez hecho esto, el programa se encarga de efectuar todos los manejos de archivos que sean necesarios de acuerdo con la rutina seleccionada.

Una vez que se hace esto, DBASE I regresa al programa principal con un código de retorno en la variable CR%. Si el valor del mismo es 0, significa que la operación fue un éxito. Si es lo 2, dependiendo del caso, nos indicará que la operación no pudo ser llevada a cabo por algún motivo.

FIGURA 2



Veamos con más detalle cada una de las operaciones disponibles:

I.-Establecer la estructura de datos. Carga los punteros del archivo índice al v ector PT% y lee las claves del campo KE$ al vector correspondiente. Esta rutina debe ser llamada antes de que cualquier operación sea realizada. Si las variables FI$ o MX% no fueron definidas con anterioridad, CR% vuelve con el valor I.

Parámetros: VV%=I, FI$=nombre del archivo índice, MX%=máximo número de registros .

2.-Agregar un registro. Se agrega un registro al archivo de datos. El índice es automáticamente actualizado para contener la clave en la posición correspondiente. Si la clave ya existía, CR% volverá con una I. Parámetros: VV%=2, A$=clave del registro a almacenar.

3.-Reescribir registro. Permite modificar uno o más campos de un registro, reescribiéndolo en la misma posición que ocupaba antes. La clave no puede ser modificada.

Parámetros: VV%=3, A$=clave del reg. a reescribir.

4.-Borrar un registro: Por medio de su

clave, elimina el registro del archivo en forma lógica (no física). Se establece un puntero indicando que el espacio ocupado por este registro puede ser utilizado para uno nuevo.
Parámetros: VV%=4, A$=clave del reg. a borrar.

5.-Leer un registro por clave nos permite acceder en forma directa a un registro cualquiera. Una vez que DBASE I le devuelve el control al programa principal, nos da en la variable NX% la posición del registro en el archivo de datos de acuerdo con el valor de su clave.
Parámetros: VV%=5, A$=clave del reg. a leer, NX%=posic. relativa.

6.-Leer registro por secuencia: nos permite leer un registro de acuerdo con su posición (ya no su clave) en el archivo. Por ejemplo, para leer el registro de clave más baja del archivo, le damos a NX% el valor I y llamamos al DBASE I con VV =6.
Parámetros: VV%=6, NX%=posic. relativa del registro a leer.

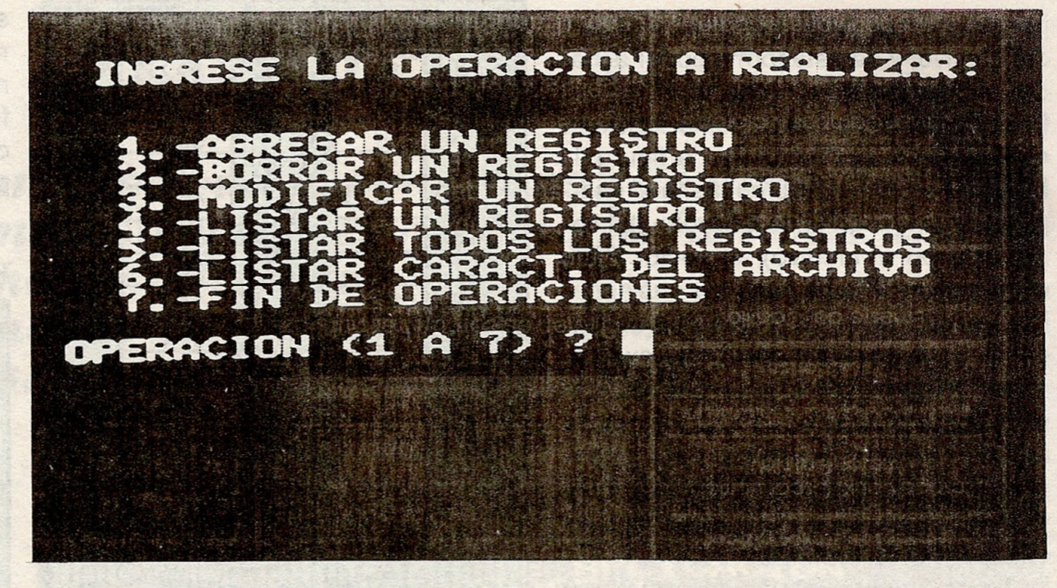
7.-Almacenar el índice: guarda los valores contenidos en PT% nuevamente en el archivo índice del disco. Esta operación debe realizarse antes de terminar las tareas, ya que en caso contrario toda la información de PT% se perdería, y los cambios efectuados al archivo no serían guardados.
Parámetros: VV%=7

8.-Características del archivo: nos da la siguiente información:
- Registros en el archivo
- Registros borrados
- Clave más baja
- Clave más alta

Parámetros: VV%=8

PROGRAMA DE EJEMPLO

Para mostrar el uso de las rutinas del DBASE I, vamos a considerar el siguiente programa que nos servirá de ejemplo. En base al mismo, cualquier lector podrá determinar su propia aplicación, ya que es una tarea realmente sencilla.



El listado consta de tres secciones. La primera es el ejemplo de uso del DBASE I , y ocupa de las líneas 1 a 502.

Luego viene el código correspondiente al DBASE I en sí, que ocupa de las líneas 1995 a 3000. Finalmente, hay una pequeña rutina para crear un archivo índice.

Los detalles que se deben tener en cuenta al usar el DBASE I, son los siguientes:

-El archivo de datos debe ser abierto como número 2.

-La variable KY$ debe ser utilizada como clave.

-Las variables FI$ y MX% deben ser establecidas como el nombre del archivo índice y la cantidad máxima de registros correspondientes.

-Antes de comenzar a trabajar con el archivo (agregar, modificar, etcétera.) se debe llamar al DBASE I con VV%=I para cargar la estructura de datos.

Cuatro variables deben ser reservadas para uso del DBASE I. Estas son KE$, PT%, PT$, y ZZ%.

Para restringir lo menos posible al usuario, el resto de las variables utilizadas por DBASE I son almacenadas en el vector ZZ%, en la línea 2009.

El programa mantiene un formato modular que es muy sencillo de entender.

Para llamar al DBASE I, se debe dar el valor correspondiente a la variable VV%, y ejecutar un GOSUB 2000. En la línea 2006, el control se pasa a la rutina correspondiente por medio de una instrucción tipo ON.. GOSUB. La variable de retorno CR% nos indica el resultado de la operación.

El programa está autodocumentado mediante REMs, y las variables importantes fueron las mencionadas en el texto.

ESTRUCTURA

A continuación, presentamos una descripción sencilla del programa según sus números de línea.

1 a 7: establecen la presentación

10 a 15: abren al archivo "LECTORES.DAT" y establecen los campos del

TABLA I

VV%	OPERACION	F1$	A$	NX%	CR%
1	Establecer estructura de datos.	Nombre de archivo índice.			0-OK 1-Nombre Indice o MX% no definido
2	Agregar registro.		Valor clave.		0-OK 1-Clave existe. 2-Archivo lleno.
3	Reescribir registro.		Clave del reg. a rees.		0-OK 1-Clave inexistente.
4	Borrar registro		Clave a reg. a borrar		0-OK 1-Clave inexistente
5	Leer reg. por clave		Valor clave	Posic. Rel. Registro	0-OK 1-Clave inexistente
6	Leer reg. por posic. relativa			Posición relativa	0-OK 1-Posición inválida
7	Almacenar índice				0-OK
8	Características del arch.				0-OK

mismo.
25: se dan los valores a MX% y FI$
32-33: menú de opciones (altas, bajas, modificaciones, visualizar, característi-cas, fin)
100-120: ingreso de datos para alta de suscriptores.
150-158: baja de suscriptores.
190-193: listar un registro (todos sus campos).
200-210: listar todos los registros (solo los nombres).
250-270: modificar un registro.
500-502: subrutina de demora.
550-554: subrutina de completar la clave hasta 6 dígitos.
2000-2010: almacenamiento de variables locales y llamada a la subrutina correspondiente.
2035-2064: subrutina que almacena el contenido del archivo índice en el vector PT% y lee las claves para almacenarlas en KE$.
2080-2088: subrutina para agregar un nuevo registro al archivo. Llama a la rutina para insertar un nuevo puntero, y localizar espacio libre en el archivo.
2090-2098= subrutina para reescribir un registro.
2100-2110= subrutina para borrar un registro. El puntero al registro borrado es almacenado en el lugar de PT% correspondiente (al final del mismo).
2150-2155= rutina para localizar un registro cuya clave esté contenida en la variable A$.
2200-2205= rutina para localizar un registro en base a su posición relativa en el archivo, por medio de la variable NX%.
2250-2259= subrutina que reescribe en el archivo índice los punteros contenidos en el vector PT%.
2280-2290= rutina para mostrar las características del archivo.
2500-2510= rutina de búsqueda dentro del vector PT%.
2520-2530= rutina para encontrar el próximo espacio libre en el disco. Si existe espacio disponible por medio de un registro borrado, se utiliza el mismo. En caso contrario, se va al final del archivo.
2540-2550= inserta el puntero a un nuevo registro en el vector PT%. El puntero es almacenado en la posición PT(K%), y todos los punteros que le siguen son corridos una posición hacia abajo.
3000-3030= subprograma para crear la estructura del archivo índice. Debe ser utilizado en cada oportunidad que trabajemos por primera vez con el DBASE I, una vez que el archivo fue creado en el disco no es necesario hacerlo más.
Las aplicaciones de estas rutinas sólo están limitadas por la imaginación del usuario. En nuestro ejemplo hemos desarrollado un archivo de suscriptores a una revista, y como pueden ver el código necesario para el mismo sólo se limita a unos cuantos INPUT'S y a las llamadas correspondientes al DBASE I. Lo único que se debe hacer para lograr otra aplicación, es copiar la parte de código que va de 2000 en adelante, y tomar como ejemplo el programa de archivo de suscriptores que yo he mandado.

Adrían D. Neme

```
1 CLS: COLOR 15,0
2 PRINT:PRINT"^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^":PRINT"          RUTINA DE AP
LICACION      ":PRINT"                  DEL DBASE 1":PRINT:PRINT"^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^^^^^^^^^^^^"
3 GOSUB 500
4 LOCATE 0,21:PRINT"                                          ":LOCATE 10,9:PRINT" ES
PERE UN MOMENTO"
7 MAXFILES=3
10 OPEN "A:ARCHIVO.DAT" AS #2 LEN=128
15 FIELD #2,6 AS KY$,20 AS NO$,6 AS FE$,10 AS OC$,15 AS CO$,22 AS DI$,20 AS CI$,
15 AS PR$,4 AS CP$
17 REM
18 REM  KY$=NUMERO DE SUSCRIPTOR  NO$=NOMBRE  FE$=FECHA NAC.
19 REM  OC$= OCUPACION  CO$=COMPUTADORA  DI$=DIRECCION
20 REM  CI$=CIUDAD  PR$=PROVINCIA  CP$=CODIGO POSTAL
21 REM
22 REM
25 MX%=500: F1$="CLAVE.IND"
30 VV%=1 : GOSUB 2000
31 REM]
32 Q%=0: CLS: PRINT" INGRESE LA OPERACION A REALIZAR:":PRINT:PRINT:PRINT"  1.-AG
REGAR UN REGISTRO":PRINT"  2.-BORRAR UN REGISTRO":PRINT"  3.-MODIFICAR UN REGIST
RO":PRINT"  4.-LISTAR UN REGISTRO":PRINT"  5.-LISTAR TODOS LOS REGISTROS"
33 PRINT"  6.-LISTAR CARACT. DEL ARCHIVO":PRINT"  7.-FIN DE OPERACIONES":PRINT:
INPUT"OPERACION (1 A 7) ";Q%
34 IF Q%=1 THEN GOSUB 100: GOTO 32
35 IF Q%=2 THEN GOSUB 150: GOTO 32
36 IF Q%=3 THEN GOSUB 250: GOTO 32
37 IF Q%=4 THEN GOSUB 180: GOTO 32
38 IF Q%=5 THEN GOSUB 200: GOTO 32
39 IF Q%=6 THEN VV%=8: GOSUB 2000: GOTO 32
40 IF Q%<>7 THEN GOTO 32
41 CLOSE: END
97 REM
98 REM
99 REM
100 CLS:PRINT"  %%%  ALTA DE SUSCRIPTORES  %%%":PRINT:PRINT:INPUT "INGRESE NRO.
DE SUSCRIPCION (FIN PARA TERMINAR) ";A$: IF A$="FIN"OR A$="fin" THEN 120 ELSE IF
 LEN (A$)>6 GOTO 100 ELSEGOSUB 550
102 VV%=5: GOSUB 2000: IF CR%<>0 THEN LSET KY$=A$: GOTO 103 ELSE PRINT"** ERROR,
 EL NRO. DE SUSCRIPTOR EXISTE":GOSUB 500:GOTO 100
103 INPUT "NOMBRE: ";F$:LSET NO$=F$
104 INPUT "FECHA NAC.:";F$:LSET FE$=F$
106 INPUT "OCUPACION :";F$: LSET OC$=F$
108 INPUT "COMPUTADORA :";F$:LSET CO$=F$
110 INPUT "DIRECCION :";F$:LSET DI$=F$
112 INPUT "CIUDAD";F$: LSET CI$=F$
114 INPUT"PROVINCIA :";F$:LSET PR$=F$
116 INPUT"CODIGO POSTAL :";F$: LSET CP$=F$
117 VV%=2: GOSUB 2000
118 IF CR%=0 THEN 100 ELSE PRINT"** ERROR-REGISTRO NO ARCHIVABLE":GOSUB 500: GOT
O 100
120 VV%=7: GOSUB 2000
122 RETURN
147 REM
148 REM
149 REM
150 ST%=0
151 CLS:PRINT" %%% BAJA DE SUSCRIPTORES %%%":PRINT:PRINT: INPUT "INGRESE EL CODI
GO DEL LECTOR A        BORRAR (FIN PARA TERMINAR):";A$: IF A$="FIN" OR A$="fin"TH
EN 156 ELSE IF LEN (A$)>6 GOTO 151 ELSE GOSUB 550
152 VV%=4: GOSUB 2000
154 IF CR%=0 THEN ST%=1 ELSE PRINT"** ERROR - EL NRO. SUSCRIPTOR NO EXISTE":GOSU
B 500: GOTO 151
155 IF ST%<>0THEN PRINT:PRINT:PRINT"                    O.K.":PRINT"EL REGISTRO YA FUE
 BORRADO DEL          ARCHIVO": GOSUB 500:GOTO 151
156 IF ST%=1 THEN VV%=7: GOSUB 2000
158 RETURN
177 REM
178 REM
179 REM
180 CLS:PRINT" %%% LISTAR UN REGISTRO %%%":PRINT:PRINT
181 INPUT" INGRESE NRO. LECTOR (FIN PARA           TERMINAR):";A$: IF A$="FIN"OR A$=
"fin" THEN 193 ELSE IF LEN (A$)>6 GOTO 181 ELSE GOSUB 550
182 VV%=5: GOSUB 2000: IF CR%<>0 THENPRINT:PRINT: PRINT"** ERROR - EL NRO. SUSCR
IPTOR NO EXISTE":GOSUB 500: GOTO 180
183 PRINT" "
184 PRINT"NOMBRE       :";NO$
185 PRINT"FECHA DE NAC.:";FE$
186 PRINT"OCUPACION    :";OC$
187 PRINT"COMPUTADORA  :";CO$
188 PRINT"DIRECCION    :";DI$
```

```
189 PRINT"CIUDAD        :";CI$
190 PRINT"PROVINCIA     :";PR$
191 PRINT"C. POSTAL     :";CP$
192 PRINT"":GOSUB 500: GOTO 180
193 RETURN
197 REM
198 REM
199 REM
200 CLS:PRINT:PRINT"  %%% LISTAR TODOS LOS REG. %%%":PRINT:PRINT: NX%=0: VV%=6:
K%=0
202 NX%=NX%+1: GOSUB 2000
204 IF CR%<>0 THEN 209
205 PRINT KY$,NO$
206 K%=K%+1: IF K%<10 THEN 202 ELSE INPUT "FIN PARA TERINAR>";Q$
207 IF Q$<>"FIN" OR Q$<>"fin" THEN K%=0: GOTO 202
209 GOSUB 500
210 RETURN
247 REM
248 REM
249 REM
250 CLS:PRINT:PRINT" %%% MODIFICACION DE UN REGISRO %%%":PRINT:PRINT
251 INPUT "INGRESE EL NRO. SUSCRIPTOR (FIN      PARA TERMINAR) :";A$: IF A$="FIN
" OR A$="fin" THEN 270 ELSE IF LEN (A$)>6 GOTO 251 ELSE GOSUB 550
252 VV%=5: GOSUB 2000
253 F$=""
254 IF CR%=1 THEN PRINT"** ERROR - EL REGISTRO NO EXISTE":GOSUB 500: GOTO 250
255 PRINT"NOMBRE: /";NO$;"/";: INPUT F$: IF LEN(F$)<>0 THEN LSET NO$=F$
256 PRINT LEN(F$):GOSUB 500
257 PRINT"FECHA NAC.: /";FE$;"/";:INPUT F$: IF LEN(F$)<>0 THEN LSET FE$=F$
258 PRINT"OCUPACION: /";OC$;"/";: INPUT F$: IF LEN (F$)<>0 THEN LSET OC$=F$
259 PRINT"COMPUTADORA: /";CO$;"/";:INPUT F$: IF LEN (F$)<>0 THEN LSET CO$=F$
260 PRINT"DIRECCION: /";DI$;"/";:INPUT F$:IF LEN (F$)<>0 THEN LSET DI$=F$
261 PRINT"CIUDAD: /";CI$;"/";:INPUT F$: IF LEN (F$)<>0 THEN LSET CI$=F$
262 PRINT"PROVINCIA: /";PR$;"/";: INPUT F$: IF LEN (F$)<>0 THEN LSET PR$=F$
263 PRINT"C. POSTAL: /";CP$;"/";: INPUT F$: IF LEN(F$)<>0 THEN LSET CP$=F$
266 VV%=3: GOSUB 2000
267 A$="FIN"
268 PRINT" ": GOTO 250
270 RETURN
500 LOCATE 0,22:PRINT"PULSE CUALQUIER TECLA PARA CONTINUAR"
501 IF INKEY$="" THEN 501
502 RETURN
550 IF LEN (A$)=6 THEN 560
552 VAR%=LEN(A$)
554 FOR J%=0 TO 5-VAR%:LET A$="0"+A$:NEXT J%
560 RETURN
1995 REM
1996 REM---------------------------------------------------------------------
1997 REM-           DBASE I (VV%,MX%,F1$,A$,PT%,PT$,NX%,CR%)                 -
1998 REM---------------------------------------------------------------------
1999 REM
2000 CR%=0:IF VV%<1 OR VV%>8 THEN CR%=1: RETURN
2001 IF VV%=1 THEN 2006:
2004 ZZ%(1)=J%:ZZ%(2)=JJ%:ZZ%(3)=K%:ZZ%(4)=LO%:ZZ%(5)=HI%:ZZ%(6)=Z%
2005 REM
2006 ON VV% GOSUB 2035,2080,2090,2100,2150,2200,2250,2280
2007 REM
2008 IF VV%=1 THEN 2010
2009 J%=ZZ%(1):JJ%=ZZ%(2): K%=ZZ%(3):LO%=ZZ%(4):HI%=ZZ%(5):Z%=ZZ%(6)
2010 RETURN
2034 REM SUBRUTINA 1:(MX%,F1$)--CARGAR PUNTERO Y CLAVES
2035 IF MX%<1 THEN CR%=1:RETURN
2037 MR%=(INT((MX%+2)/64)+1)*64
2038 DIM PT$(64),PT%(MR%),KE$(MX%),ZZ%(8)
2040 OPEN F1$ AS #1 LEN=128 :REM ARCHIVO INDICE
2042 FOR J%=1 TO 64:FIELD #1,(J%-1)*2 AS DU$, 2 AS PT$(J%): NEXT J%
2050 K%=0: IF LOF(1)=0 THEN 2062
2051 FOR J%=1 TO INT(MR%/64)
2052 GET 1,J%
2054 FOR JJ%=1 TO 64: K%=K%+1: PT%(K%)=CVI(PT$(JJ%)):NEXT JJ%
2055 NEXT J%
2056 REM
2057 IF PT%(MR%)=0 THEN 2062
2058 FOR J%=1 TO PT%(MR%)+PT%(MR%-1)
2059 GET 2,J%:KE$(J%)=KY$
2060 NEXT J%
2062 RETURN
2079 REM SUBRUTINA 2:(MR%,A$,CR%)--AGREGAR UN REGISTRO
2080 GOSUB 2500:IF K%>0 THEN CR%=1: GOTO 2088
2083 GOSUB 2520: IF Z%>MR%-1 THEN CR%=2: GOTO 2088
2085 K%=-K%:GOSUB 2540
2086 KE$(Z%)=A$
2087 PUT 2,Z%
2088 RETURN
2089 REM SUBRUTINA 3:(REESCRIBIR REGISTRO)
2090 GOSUB 2500: IF K%<0 THEN CR%=1: GOTO 2098
2092 PUT 2,PT%(K%)
2098 RETURN
2099 REM SUBRUTINA 4:(MR%,A$,CR%)---BORRAR UN REGISTRO
2100 GOSUB 2500: IF K%<0 THEN CR%=1: GOTO 2110
2102 Z%=PT%(K%): IF K%=PT%(MR%) THEN 2107
2104 FOR J%=K% TO PT%(MR%)-1: PT%(J%)=PT%(J%+1): NEXT J%
2107 JJ%=PT%(MR%-1)
2108 PT%(PT%(MR%))=0: PT%(MR%)=PT%(MR%)-1:PT%(MR%-1)=JJ%+1:PT%(MR%-2-JJ%)=Z%
2110 RETURN
2149 REM SUBRUTINA 5:(MR%,NX%,CR%)-- LEER REGISTRO POR CLAVE
2150 GOSUB 2500: IF K%<0 THEN CR%=1: GOTO 2155
2152 GET 2,PT%(K%)
2153 NX%=K%
2155 RETURN
2199 REM SUBRUTINA 6:(MR%,NX%,CR%)---LEER REGISTROS
2200 IF NX%<0 OR NX%>PT %(MR%) THEN CR%=1: GOTO 2205
2203 GET 2,PT%(NX%)
2205 RETURN
2249 REM SUBRUTINA 7:(MR%)---GUARDA PUNTEROS
2250 K%=0: Z%=INT((PT%(MR%)-1)/64)+1
2252 FOR J%=1 TO Z%
2253 FOR JJ%=1 TO 64: K%=K%+1: LSET PT$(JJ%)=MKI$(PT%(K%)): NEXT JJ%: PUT 1,J%
2254 NEXT J%
2255 K%=INT(MR%/64): IF Z%=K% THEN 2259
2257 K%=(K%-1)*64: FOR J%=1 TO 64: LSET PT$(J%)=MKI$(PT%(J%+K%)): NEXT J%: PUT 1
, INT (MR%/64)
2259 RETURN
2279 REM SUBRUTINA 8: ESTADISTICAS DEL ARCHIVO
2280 CLS:PRINT:PRINT:PRINT" %%% ESTADISTICAS DEL ARCHIVO %%%":PRINT:PRINT
2281 PRINT" ":IF PT%(MR%)=0 THEN PRINT"** NO HAY REGISTROS EN EL ARCHIVO **":FOR
 VAR%=1 TO 500: NEXT VAR%: GOTO 2290
2283 PRINT" REGISTROS EN ARCHIVO: ";PT%(MR%)
2284 PRINT" REGISTROS BORRADOS  : ";PT%(MR%-1)
2285 PRINT" CLAVE MAS BAJA      : ";KE$(PT%(1))
2286 PRINT" CLAVE MAS ALTA      : ";KE$(PT%(PT%(MR%)))
2287 PRINT" "
2289 GOSUB 500
2290 RETURN
2498 REM
2499 REM SUBRUTINA DE BUSQUEDA BINARIA
2500 IF PT%(MR%)=0 THEN K%=-1: RETURN
2502 LO%=0: HI%=PT%(MR%)+1
2504 M%= INT((LO%+HI%)/2)
2505 IF A$=KE$(PT%(M%)) THEN K%=M%: GOTO 2510
2506 IF A$> KE$(PT%(M%)) THEN LO%=M% ELSE HI%=M%
2508 IF LO%+1<>HI% THEN 2504 ELSE K%=-HI%
2510 RETURN
2518 REM
2519 REM SUBRUTINA PARA ENCONTRAR ESPACIO LIBRE EN EL ARCHIVO
2520 IF PT%(MR%-1)=0 THEN Z%=PT%(MR%)+1: GOTO 2530
2522 J%=PT%(MR%): JJ%=PT%(MR%-1)
2524 Z%=PT%(MR%-1-JJ%): PT%(MR%-1)=PT%(MR%-1)-1: PT%(MR%-1-JJ%)=0
2530 RETURN
2538 REM
2540 IF K%=PT%(MR%)+1 THEN 2548
2542 FOR J%=PT%(MR%)+1 TO K%+1 STEP -1
2544 PT%(J%)=PT%(J%-1)
2545 NEXT J%
2548 PT%(K%)=Z%: PT%(MR%)=PT%(MR%)+1
2550 RETURN
2997 REM
2998 REM RUTINA DE INICIO DE ARCHIVO INDICE
2999 REM
3000 PRINT" ":PRINT " CREANDO ARCHIVO INDICE":PRINT" "
3001 INPUT "DRIVE PARA CONTENER DATOS"; UA$
3002 INPUT "NOMBRE DEL ARCHIVO ";F$
3004 INPUT "MAXIMO NUMERO DE REGISTROS";MX%
3006 MR%=(INT((MX%+2)/64)+1)*64
3008 DIM PT$(64)
3009 REM ABRIR ARCHIVO Y CREAR PUNTEROS
3010 OPEN F$ AS #1 LEN=128
3012 FOR J%=1 TO 64: FIELD #1,(J%-1)*2 AS DU$,2 AS PT$(J%):NEXT J%
3014 ZR$=MKI$(0):FOR J%=1 TO 64: LSET PT$(J%)=ZR$: NEXT J%
3015 REM ALMACENAR BLOQUES DE INDICE 0
3016 FOR J%=1 TO MR%/64
3018 PUT 1,J%
3020 NEXT J%
3022 PRINT" ":PRINT" INICIALIZACION COMPLETA EN DRIVE ";UA$
```

CONCURSO MENSUAL

MSX-TEST

CONCURSO MENSUAL

PREMIO



Un soft a eleccion entre IDEA BASE, IDEA TEXT, o BASIC TUTOR.

Auspiciado por TELEMATICA S.A. fabricante en Argentina de las computadoras personales TALENT MSX

Para participar en este certamen deben señalar cual es la información correcta que presenta cada ítem. Junto con las respuestas deben remitir los datos en el correspondiente cupón. Los premios se entregarán en la administración de la editorial. Los que no puedan concurrir pueden solicitar el envio pagando el franqueo contrareembolso. Los premios podrán ser reclamados dentro de los 120 días después de haber sido anunciados.

CIERRE 1º DE MAYO DE 1988

3

1) ¿Quién creó el vocablo "robot" para designar a los aparatos mecánicos que realizan algún trabajo?
a) Julio Verne
b) Aldous Huxley
c) Karel Capek

2) ¿Qué significa la sigla PSG?
a) Packaged software of Germany
b) Planning systen gate
c) Programable sound generator

3) El byte equivale a
a) 4 bits
b) 6 bits
c) 8 bits

4) ¿A qué tipo de juegos se denomina ARCADE?
a) A los mata-marcianos
b) A los de estrategia
c) A las carreras de motos

5) ¿Quiénes son los diseñadores del brazo robot RBS-1?
a) Caro, Tokuda y Graffigna
b) Bauer y Feito
c) Delfino, Romero y Veloso

Nombre y apellido____________________

Dirección:____________________

Documento:__________ Edad:_____ Máquina:__________

Qué es lo que más me gusta de la revista:____________________

Que le agregaría:____________________

Que es lo que no me gusta:____________________

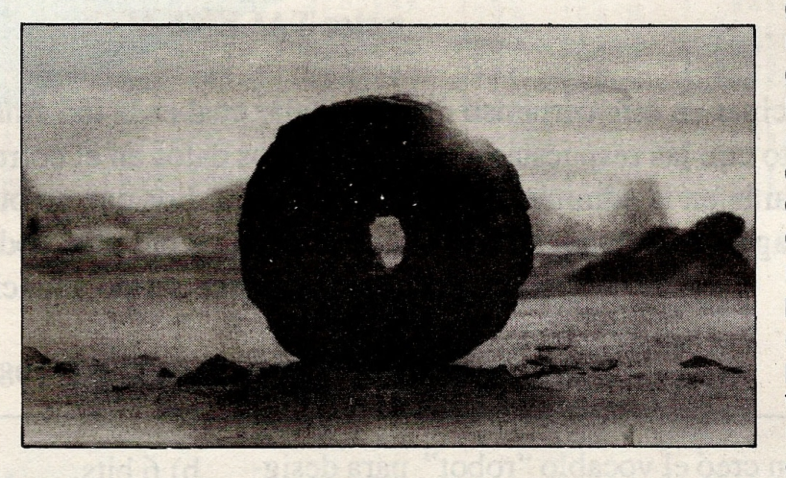
LOS CICLOS WHILE ... DO ...

La realzación de una tarea cualquiera por medio de un sistema informatizado incluye una serie de pasos, entre los cuales la ejecución del programa codificado en BASIC no es sino el último de ellos.

Los principales pasos para la realización de un programa son:
a) la enunciación clara, simple y completa de la tarea.
b) el análisis de la misma y la elaboración del algoritmo apropiado. c) la confección del diagrama de flujo correspondiente.
d) la codificación del algoritmo en un lenguaje de programación.
Ver figura 1.
En este artículo trataremos el tema relativo a las estructuras cíclicas denominadas DO ... WHILE ..., su empleo, la diagramación de las mismas y su codificación en BASIC.

ANALISIS

Para un usuario de microcomputadoras que acostumbra a efectuar sus propios programas en BASIC, la única sugerencia que podemos dar, enmarcada en los objetivos de difusión de la revista, es recomendar que se divida el proyecto global en tantas partes pequeñas como sea posible; a éstas las llamaremos "MODULOS".
Dichos módulos, cuando han sido correctamente limitados, son mucho más fáciles de resolver por su menor complejidad estructural. La modularización no debe culminar en el proceso de análisis, sino que debe extenderse a la diagramación, la comprobación de escritorio, la codificación en BASIC y la depuración ulterior.
El resultado del proceso de análisis, secuencia ordenada de pasos que resolverán nuestro problema, se denomina ALGORITMO; y si lo tradujéramos libremente del árabe, lo llamaríamos "receta".



DIAGRAMACION

A la hora de convertir el algoritmo en un gráfico que sea más simple de digerir, el DIAGRAMA, y que a la vez nos facilite el proceso posterior de codificación, muchos programadores se ponen a practicar el salto en alto, y salvan limpiamente el obstáculo, pasando directamente a la codificación.
En realidad, quienes suelen saltear la etapa de diagramación, ya que están, se evitan también la del análisis previo, ...total, ¡¿para qué...?!, y se ponen a "programar" directamente, con unos resultados que para qué le voy a contar...
En nuestras notas sobre el tema convengamos en utilizar los bloques que se muestran en la Figura 2 y en seguir las siguientes reglas prácticas:
1 - Emplear un bloque para cada sentencia BASIC, salvo las REM.
2 - Salir de los bloques de decisión hacia la derecha por el SI (CIERTO)
3 - Numerar en forma lineal descendente, y de izquierda a derecha.
4 - Asignar a todas las líneas de interconexión acodadas una bifurcación incondicional (GO TO).
5 - Disponer el diagrama de cada módulo sobre el margen izquierdo de la hoja y colocar su codificación correspondiente.
6 - Procurar el apareamiento BLOQUE (diagrama) - SENTENCIA (programa), en la misma línea.

FIGURA 1

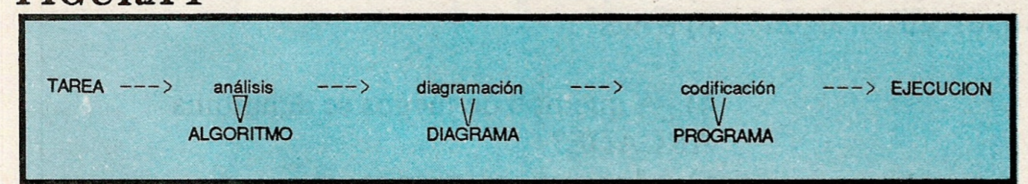
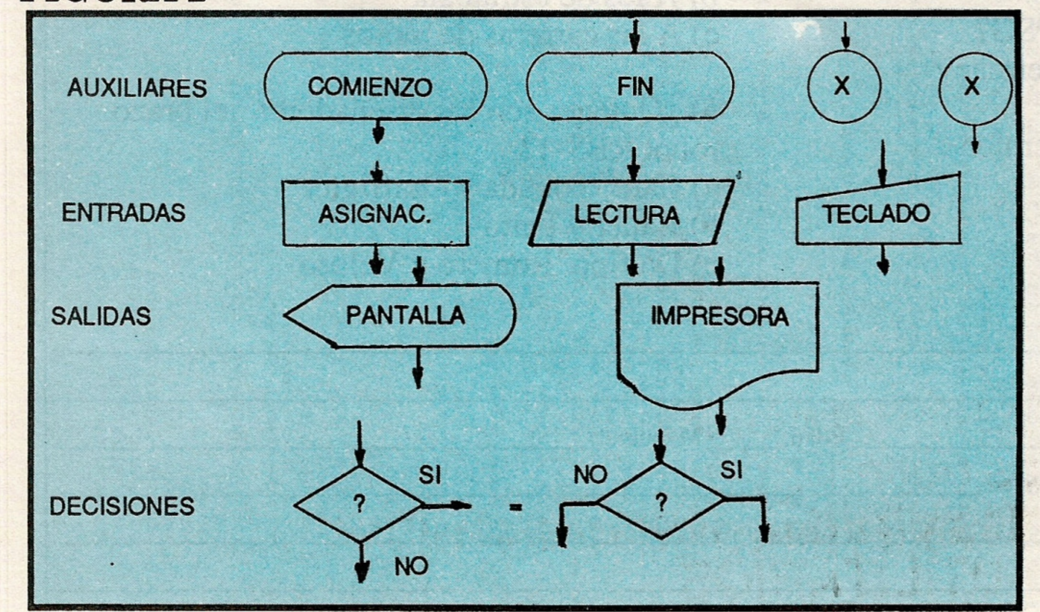
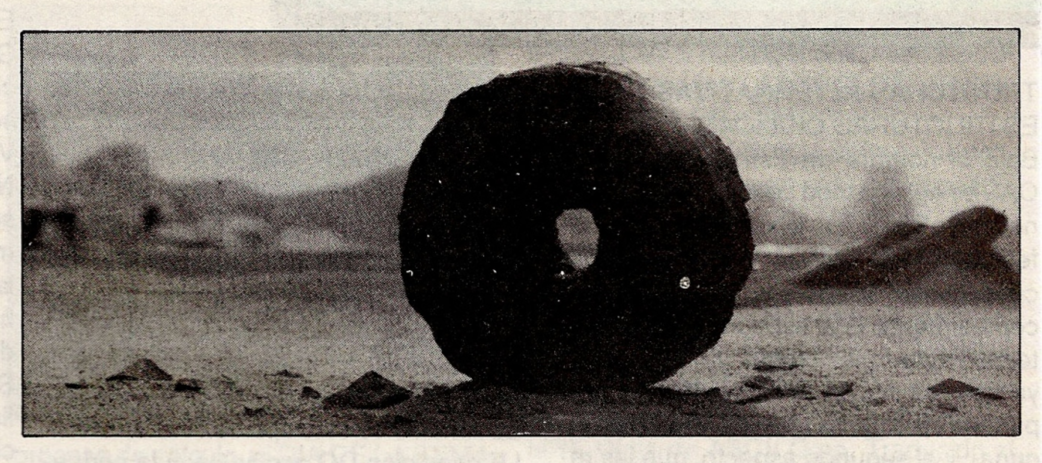


FIGURA 2



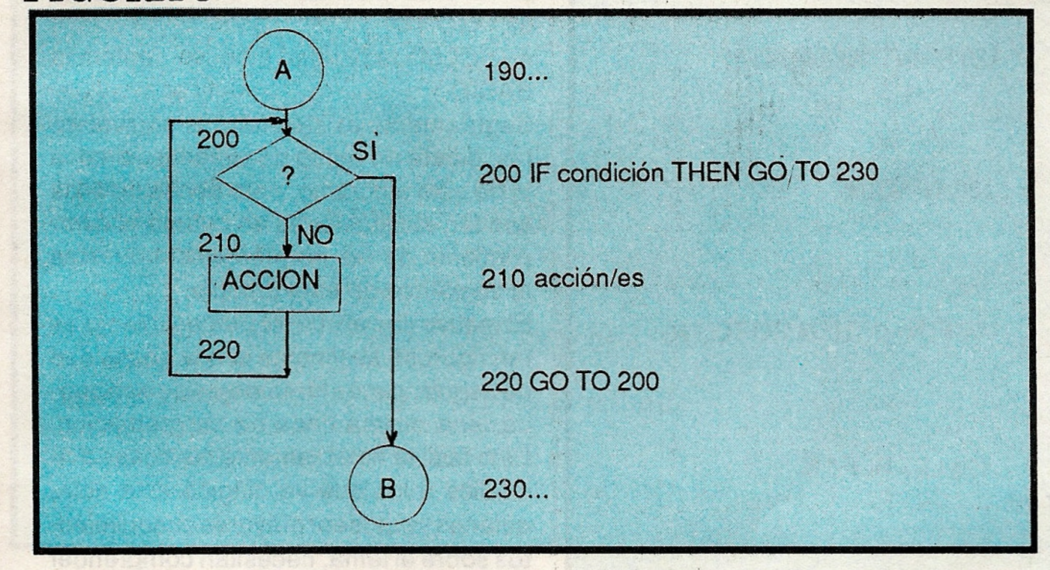
Mencionemos que somos fervientes partidarios de la diagramación lógica LINEAL vs la ESTRUCTURADA ya que el BASIC es esencialmente un lenguaje LINEAL, casi nada estructurado, y porque, además, la práctica docente demuestra que a los adultos les resulta mucho más fácil de comprender y sencillo de codificar que otros, dada la relación que hay entre cada bloque y la sentencia BASIC que le corresponde.
Ver figura 2



CODIFICACION

Codificar significa traducir el algoritmo, una vez llevado al DIAGRAMA DE FLUJO, a un lenguaje de programacion que sea comprensible por el microprocesador. Lo que no debemos olvidar es que también debe ser comprensible para quien desee leerlo, y por eso resulta conveniente que este paso sea efectuado de la manera más clara y sencilla posible.
Para lograr optimizar la legibilidad de los programas en BASIC, sugerimos hacerlo pensando en nosotros mismos y no en la máquina. De ella no necesitamos encargarnos.
Conviene señalar los siguientes factores positivos:
1 - En las asignaciones, usar el LET y seleccionar los identificadores de forma tal que se relacionen fácilmente con el dato que guardan.
2 - Los ingresos de datos por teclado, INPUT, deben ir precedidos por un mensaje que indique claramente qué dato es esperado, pero a través de un PRINT, evitando mezclar funciones.
3 - Usar REMarks para señalar cada bloque o módulo con un nombre significativo.
4 - Dejar espacios entre cada elemento de las líneas.
5 - Marginar sobre la izquierda las sentencias interiores de los ciclos.
6 - Evitar las lineas multisentencias, dentro de lo posible.
En aquellos casos que se requiera mayor velocidad de ejecución, podemos recurrir a confeccionar dos versiones del mismo programa: una para nosotros, bien clara y suficientemente documentada, y otra para la máquina, donde se apunte a minimizar el tiempo de ejecución.

FIGURA 3



LAS ESTRUCTURAS DE CONTROL

Gran parte del poder que brindan la computadoras deriva del hecho de poder tomar decisiones y determinar qué rumbo seguir o qué acciones ejecutar. Esto se hace sobre la base de los datos que se ingresan por teclado, o bien por los valores resultantes de los cálculos que se efectúan internamente.
Esta es la gran diferencia entre la computadora y cualquier otra máquina que solo pueda seguir una secuencia estrictamente lineal de instrucciones, como si fuera un simple mecanismo de relojería.
Las dos principales estructuras que permiten alterar la ejecución secuencial o lineal de un programa son: a) las ES-

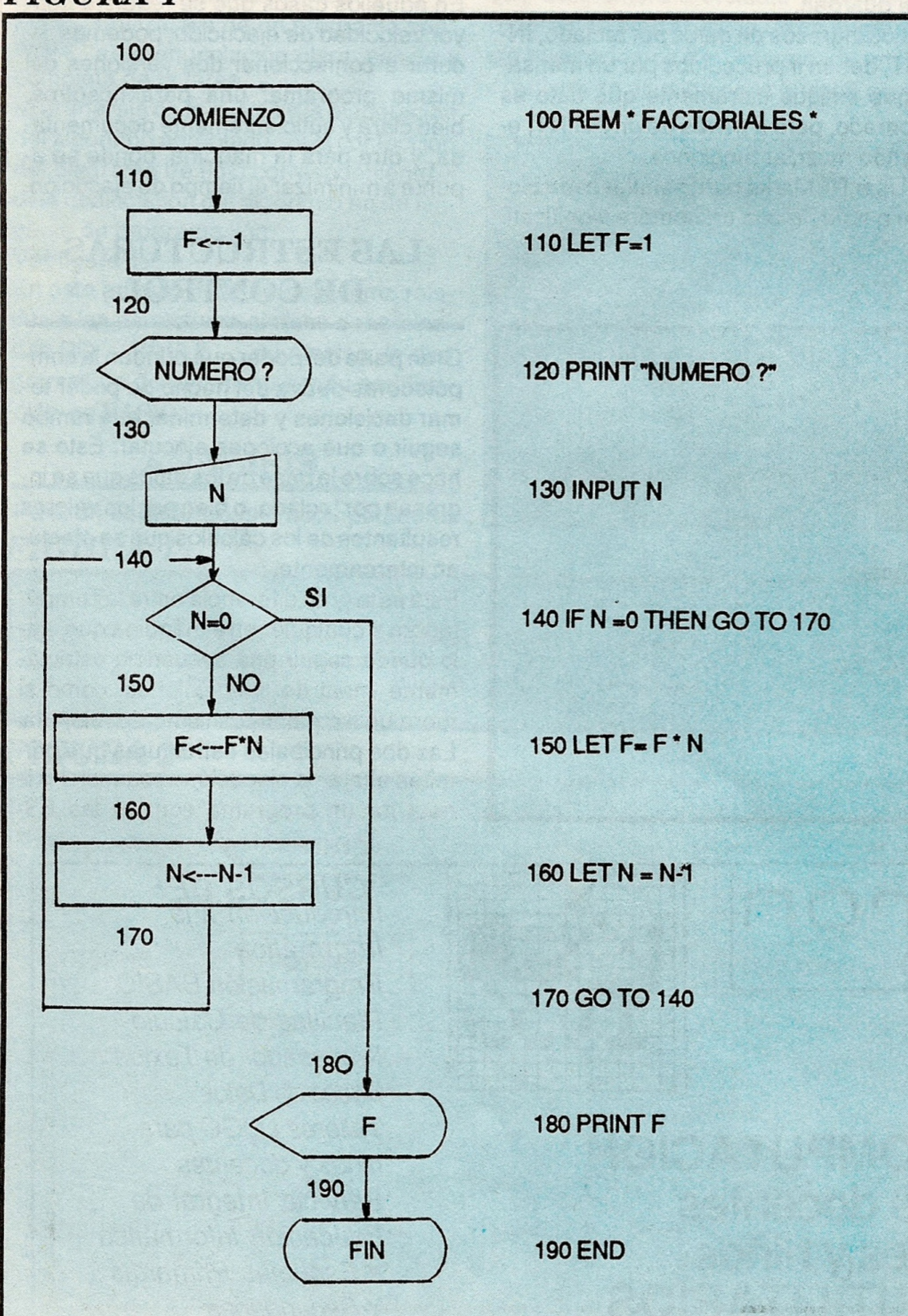
TRUCTURAS ALTERNATIVAS y b) las ESTRUCTURAS CICLICAS.
Bajo el nombre de Estructuras CICLICAS se agrupa toda secuencia de acciones cuya ejecución puede llegar a ser efectuada más de una vez, y en forma consecutiva. Debemos destacar conceptualmente dos aspectos sobre este tema: Primero, el "puede" y no el "debe", ya que las acciones inherentes al ciclo pueden ejecutarse una sola vez, ...!o ninguna!; y el segundo aspecto, que las repeticiones deben ser consecutivas, dado que si no lo fueran, se constituiría lo que denominamos SUBRUTINAS.
En este artículo solo trataremos los Ciclos conocidos como WHILE ... DO ..., que son aquellos que tienen la sentencia de control al comienzo, de modo tal que, en el caso de no cumplirse la condición que debe verificarse en ella, las acciones especificadas en el ciclo no llegan a ejecutarse ninguna vez.

CICLOS WHILE ... DO ...

La sintaxis de este tipo de ciclo es: WHILE condición DO acción/es, y la podríamos expresar como MIENTRAS (se cumpla la) condición (de reciclaje) HACER (las siguientes) acción/es.

En la figura 3 hemos preparado un DIAGRAMA DE FLUJO esquemático de un ciclo WHILE/DO, en el cual pueden observarse sus características esenciales. Ver figura 3
Nótese como el bloque de control representado por el rombo se encuentra al comienzo del ciclo. Para poder cumplir con la premisa impuesta de salir por el SI , hemos cambiado reciclaje por una condición de salida del ciclo.
Si la condición de salida se cumple, el flujo deriva hacia el módulo que sigue al ciclo, en caso contrario, se ejecutan las acciones pertenecientes al mismo .
En caso de que al llegar -por primera vez- al comienzo del ciclo la condición de salida se cumpla, las sentencias internas no son ejecutadas ninguna vez.
En la figura 4 podemos analizar un sencillo ejemplo de CALCULO DEL FACTORIAL DE UN NUMERO, definido en forma cíclica a través de un WHILE/DO.
Ver figura 4
Observemos como se verifican en este ejemplo los siguientes puntos.
1 - respeta la salida hacia la derecha por el SI.
2 - respeta la numeración descendente y de izquierda a derecha.
3 - respeta la concordancia línea acodada - GO TO.
4 - respeta la paridad bloque - sentencia BASIC.
Seguramente existen formas equivalentes, desde un punto de vista funcional, a la de este algoritmo, pero pocas de ellas son tan satisfactorias en cuanto al cumplimiento de las normas autoimpuestas al comienzo de este artículo.
El motivo por el cual el procedimiento es tan riguroso se vincula con la necesidad de seguir, lo más firmemente que se pueda, una misma línea de programación. Esto facilita notablemente las cosas a aquellos a los que va dirigida esta nota, quienes sin poseer mayores conocimientos sobre el tema, necesitan comprender con absoluta claridad la esencia de las principales estructuras, en qué casos usarlas, etcétera.

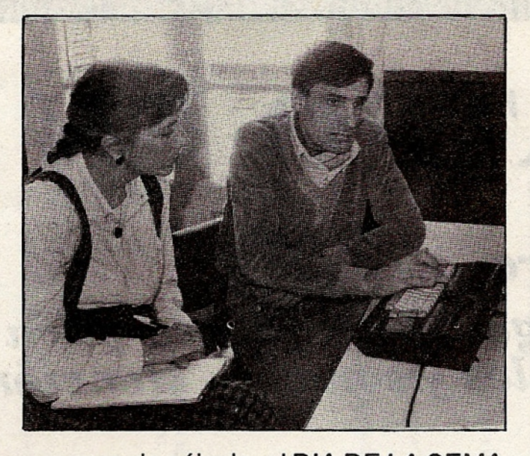
FIGURA 4



OBSERVACIONES FINALES

El caso de los ciclos WHILE/DO se impone en todos aquellos casos en que deba verificarse PREVIAMENTE la condición de salida del ciclo, como por ejemplo cuando algún dato de entrada no pueda ser 0 (CERO) para evitar que se efectúe algún cálculo que incluya productos (cualquier valor * 0 = 0) o cocientes (im-

posible dividir por 0), etcétera.
Es necesario advertir a los fanáticos del FOR ... TO .../NEXT ..., que lo usan para cualquier caso, aunque la estructura en cuestión solo se parezca a un ciclo, que tratar de encarar la ejecucción de un ciclo WHILE/DO a través de una sentencia FOR/NEXT puede ocasionar dolor de cabeza...
Para finalizar hemos agregado un programa preparado por un grupo de alumnos adolescentes, en el cual se ha empleado un CICLO WHILE/DO en el módulo de cálculo de suma de los días del año. El programa toma como dato de entrada una fecha, expresada en año, mes y día, y devuelve como resultado de un proceso de cálculo, el DIA DE LA SEMANA que cae.
Ver figura 5



Este ejemplo sirve también para trabajar los temas inherentes al manejo de datos estructurados homogéneos unidimensionales, llamados ARREGLOS UNIDIMENSIONALES o VECTORES.
NOTA: El original de este artículo fue confeccionado por el autor con un equipo SEMI-PROFESIONAL de la línea MSX, compuesto por una micro hogareña TALENT MS X DPC-200, Unidad de Discos Flexibles TALENT DPF-550, Pantalla común TV color e Impresora SEIKOSHA SP-1000AS de 100 cps y NLQ. Como procesador de Textos se ha utilizado un MSX-Write de ASCII Corp. Traducido al castellano por TELEMATICA S.A.

Gustavo O. Delfino

FIGURA 5

```
10 REM ********************
12 REM *                  *
14 REM * DIA DE LA SEMANA *
16 REM *                  *
18 REM ********************
20 DIM MES$(12),QD(12)
22 REM LECTURA MESES
24 LET M = 1
26  READ MES$(M),QD(M)
28 LET M = M + 1
30 IF M < = 12 THEN GO TO 26
32 DATA enero,31,febrero,28,marzo,31,
   abril,30,mayo,31,junio,30,julio,31,
   agosto,31,setiembre,30,octubre,31,
   noviembre,30,diciembre,31
34 REM LECTURA DIAS
36 LET D = 1
38  READ DIA$(B)
40 LET D = D + 1
42 IF D < = 7 THEN GO TO 38
44 DATA martes,miércoles,jueves,
   viernes,sábado,domingo,lunes
46 REM INGRESO DE DATOS
48 CLS:SCREEN 0:WIDTH 40:KEY OFF:
   COLOR 4,14,14
50 LOCATE 8,2:PRINT "Ingresa el año
   (01 a 99)"
52 LOCATE 8,4:INPUT AA
54 IF AA < 1 OR AA > 99 THEN GO TO 52
56 LOCATE 8,7:PRINT "Ingresa el mes
   (1 a 12)"
58 LOCATE 8,9:INPUT MM
60 IF MM < 1 OR MM > 12 THEN GO TO 58
62 LOCATE 8,12:PRINT "Ingresa el día
   (1 a 31)"
64 LOCATE 8,14:INPUT DD
66 IF DD < 1 OR DD > 31 THEN GO TO 64
68 REM CALCULO AÑO BISIESTO
70 IF AA MOD 4 = 0 THEN LET BIS = 1
   ELSE LET BIS = 0
72 REM CALCULO SUMA DIAS/AÑO
74 LET M = 1
76 IF M > MM - 1 THEN GO TO 86
78  LET S = S + QD(M)
80  IF M = 2 AND BIS = 1 THEN LET S =
    S + 1
82  LET M = M + 1
84 GO TO 76
86 LET S = S + DD
88 REM CALCULO QUE DIA CAE
90 LET SUM = (AA-1) * 365 + INT((AA-1)
   /4) + S
92 LET Z = SUM - 7 * INT(SUM/7)
94 IF Z = 0 THEN LET Z = 7
96 REM DISPLAY RESULTADO
98 LOCATE 2,17:PRINT "El";DD;"de ";
   MES$(MM);" de ";"19";MID$(STR$(AA)
   ,2,2);" cae ";DIA$(Z)
```

CUATRO EN LINEA

Clase: Juego
Autor: Alberto Rivera

Un apasionante juego de estrategia para dos jugadores, con buenos efectos visuales y sonoros. Mención del tercer concurso de programas.

Este es un juego de tablero conocido por casi todos los estudiantes y por todos los que tienen inclinaciones lúdicas.

El juego consta de un tablero de 6 filas por 7 columnas y 42 fichas, 21 de cada color.

El que logra alinear cuatro fichas de su color es el ganador.

El programa está diseñado para que se pueda jugar con joysticks o con los cursores.

Los jugadores deben colocar en el tablero, alternadamente, las fichas que les corresponden, de a una por vez. Para ello deberán hacer desplazar la flecha que aparece en la pantalla, la misma indica la columna donde depositarán su ficha. Pulsando el botón, la ficha caerá y se ubicará sobre el último lugar vacío.

Las fichas conservan ese lugar hasta que haya terminado el juego.

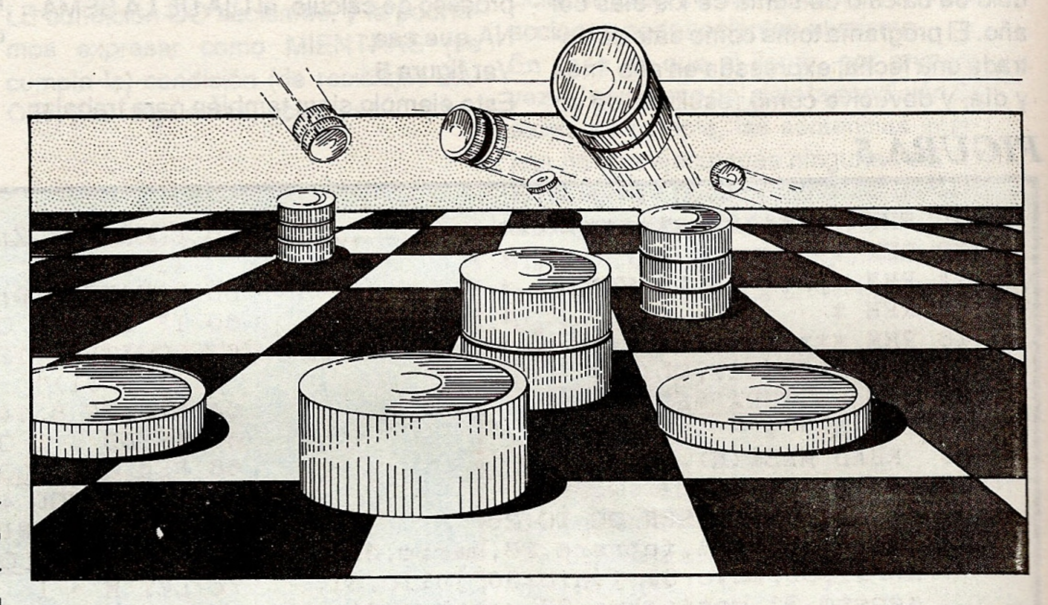


DIAGRAMA DE BLOQUES



10-160 menú del programa
170-220 presentación de instrucciones
230-280 registra opciones de manejo
290-380 comienzo del juego
390-440 programa principal de juego
450-690 gráficos en pantalla
700-850 caída de la ficha
860-910 sonidos
920-1030 movimientos del jugador
1040-1080 colocación de la ficha (pinta el color correspondiente)
1090-1180 verifica si hay cuatro fichas en línea
1190-1310 designa el ganador
1320-1340 forma sprites

VARIABLES IMPORTANTES

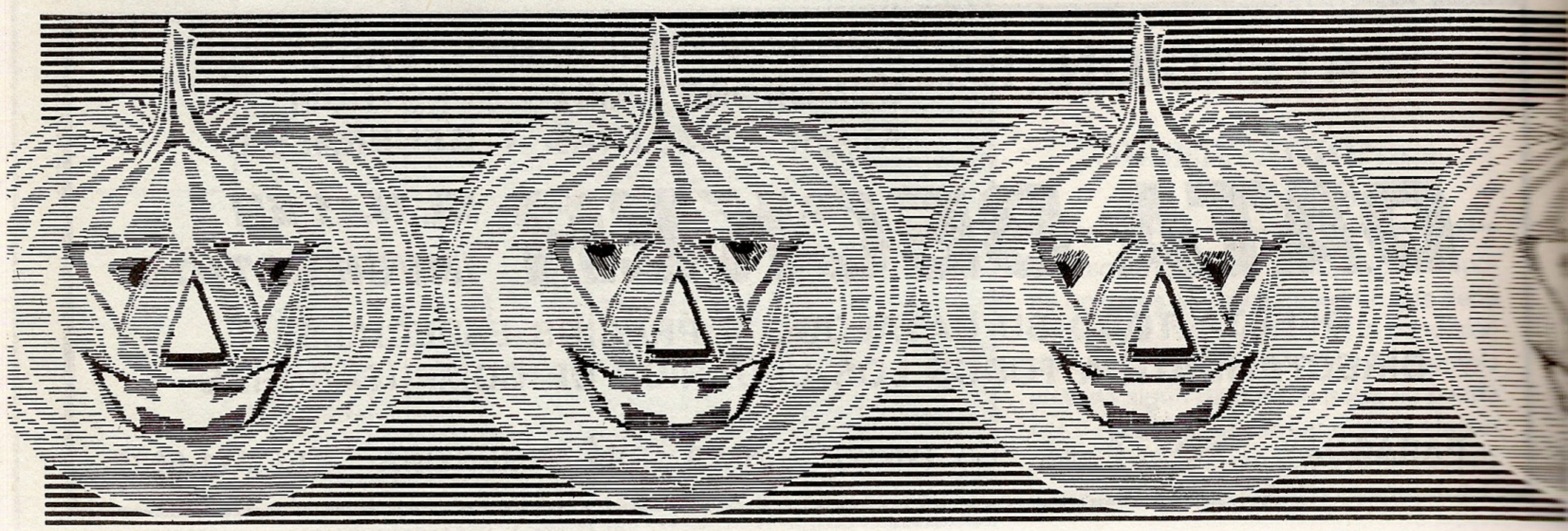
PO(x,y) verifica la existencia o no de una ficha en un determinado lugar del tablero. Toma valor 0 cuando el lugar está desocupado; toma valor 1, si el lugar está ocupado por una ficha del jugador Nº 1; y toma valor 2, si en este lugar hay depositada una ficha del jugador Nº 2.

CO: indica la columna desde la cual se lanzó la ficha.

Y: valor de la coordenada "y" donde está ubicada la última ficha depositada.

T: verifica la ocupación total del tablero, para comenzar nuevamente el juego.

X,Y,X1,Y1,X2,Y2,X3,Y3: contienen los valores de las coordenadas "x" e "y" ya alineadas.



```
10 '**********
20 '*  MENU  *
30 '**********
40 A$(1)="TECLADO":A$(2)=A$(1)
50 COLOR 15,4,4:SCREEN0:KEY OFF
60 WIDTH 29:LOCATE,4:PRINT "                MENU":PRINT
70 PRINT STRING$(27,"*"):PRINT:PRINT
80 PRINT " 1. INSTRUCCIONES":PRINT
90 PRINT " 2. JUGAR CON TECLADO":PRINT
100 PRINT " 3. JUGAR CON TECL. Y JOY.":PRINT
110 PRINT " 4. JUGAR CON JOYSTICK":PRINT:PRINT
130 PRINT STRING$(27,"*")
140 A$=INKEY$:IF A$="" THEN 140 ELSE IF ASC(A$)<49 OR ASC(A$)>53 THEN 140
150 A=VAL(A$):ON A GOSUB 170,260,270,280,300
160 IF A=1 THEN GOTO 50 ELSE 320
170 '*****************
180 '* INSTRUCCIONES *
190 '*****************
200 WIDTH 37:SCREEN 0
210 CH$=STRING$(37," "):PRINT " PARTICIPAN DOS JUGADORES , ESTOS DE-";CH$;"BERAN
 INTRODUCIR EN EL TABLERO , AL-";CH$;" TERNADAMENTE , LAS FICHAS.";CH$
211 PRINT " EL OBJETIVO DEL JUEGO ES ALINEAR    ";CH$;"CUATRO FICHAS DE UN MISMO
 COLOR EN    ";CH$;"FORMA HORIZONTAL , VERTICAL U OBLICUA";CH$;"EN CUALQUIER PART
E DEL TABLERO.";CH$
212 PRINT " EL QUE LO LOGRE PRIMERO GANA.";CH$
213 PRINT "LA FLECHA INDICA LA COLUMNA DONDE     ";CH$;"CAERA LA FICHA.";CH$
214 PRINT " EN LA PARTE INFERIOR DE LA PANTALLA ";CH$;"SE INDICA EL Nro. DEL JUG
ADOR."
220 IF INKEY$="" THEN 220 ELSE RETURN
230 '********************
240 '* OPCIONES MANEJO *
250 '********************
260 A$(1)="TECLADO":A$(2)=A$(1):A(1)=0:A(2)=0:RETURN
270 A$(1)="TECLADO":A$(2)="JOYSTICK 1":A(1)=0:A(2)=1:RETURN
280 A$(1)="JOYSTICK 1":A$(2)="JOYSTICK 2":A(1)=1:A(2)=2:RETURN
290 '***************
300 '* 4 EN LINEA *
310 '***************
320 SCREEN 2,3:OPEN"GRP:"AS#1:COLOR 15,4,4:GOSUB 870:GOSUB 460
330 '## COMIENZO DEL JUEGO ##
340 LINE(96,182)-(169,190),1,BF
350 PSET(100,183),1:COLOR15:PRINT#1,"COMIENZO!"
360 FOR I=1 TO 50:A$=INKEY$:NEXT
370 IF INKEY$="" THEN 370 ELSE BEEP
380 LINE(96,182)-(169,190),4,BF
390 '## JUEGO ##
400 DIMPO(7,6):J=2:CO=1
410 IF J=1 THEN J=2:C=2 ELSE J=1:C=15
420 GOSUB 920:'MUEVE FLECHA
430 GOSUB 870:SOUND 8,15:'PREPARA SONIDOS
440 GOSUB 710:GOTO 410:'TIRA FICHA Y VUELVE AL CICLO
450 '## GRAFICO PANTALLA ##
460 FORX=30TO255STEP30:LINE(X,0)-(X,180):NEXT
470 FORY=0TO180STEP30:LINE(30,Y)-(240,Y):NEXT
480 FORX=45TO233STEP30:FORY=15TO175STEP30:CIRCLE(X,Y),12:NEXTY,X
490 GOSUB 530:'SPRITES
500 GOSUB 550:'DIBUJO LETRAS
510 RETURN
520 '## DEFINICION SPRITES ##
530 FORI=0TO2:FORJ=1TO32:READA:S$=S$+CHR$(A):NEXT:SPRITE$(I)=S$:S$="":NEXT:RETUR
N
540  ## GRAFICO LETRAS ##
550 GOSUB 570:Y1=-1:GOSUB 590:Y1=-3:GOSUB 610:GOSUB 630:GOSUB 650:Y1=71:GOSUB610
:Y1=118:GOSUB 590:GOSUB 670:GOSUB 690:RETURN
560 '** 4 **
570 X=10:FOR Y=0 TO 10:LINE (X,Y)-(X+5,Y):LINE (10,Y)-(15,Y):X=X-1:NEXT:FOR Y=10
 TO 15:LINE (0,Y)-(17,Y):NEXT:FOR Y=15 TO 20:LINE (10,Y)-(15,Y):NEXT:RETURN
580 '** E **
590 FOR Y=28TO33:LINE (0,Y+Y1)-(15,Y+Y1):NEXT:LINE (0,34+Y1)-(5,34+Y1):LINE (0,3
5+Y1)-(5,35+Y1):FOR Y=36 TO 40:LINE (0,Y+Y1)-(12,Y+Y1):NEXT:LINE (0,41+Y1)-(5,41
+Y1):LINE (0,42+Y1)-(5,42+Y1):FOR Y=43 TO 48:LINE (0,Y+Y1)-(15,Y+Y1):NEXT:RETURN
600 '** N **
610 X=0:FOR Y=52 TO 73:LINE (0,Y+Y1)-(5,Y+Y1):LINE (X,Y+Y1)-(X+5,Y+Y1):LINE (10,
Y+Y1)-(15,Y+Y1):X=X+.5:NEXT:RETURN
620 '** L **
630 FOR Y=77 TO 91:LINE (0,Y)-(5,Y):NEXTY:FOR Y=92 TO 97:LINE (0,Y)-(15,Y):NEXT:
RETURN
640 '** I **
650 FOR Y=99 TO 104:LINE (2,Y)-(13,Y):NEXT:FOR Y=105 TO 115:LINE (5,Y)-(10,Y):NE
XT:FOR Y=116 TO 121:LINE (2,Y)-(13,Y):NEXT:RETURN
660 '** A **
670 X=5:X1=10:FOR Y=168 TO 173:LINE(X,Y)-(X1,Y):X=X-1:X1=X1+1:NEXT:LINE (0,174)-
(6,174):LINE (9,174)-(15,174):FOR Y=175 TO 177:LINE(0,Y)-(5,Y):LINE(10,Y)-(15,Y)
:NEXT:FOR Y=178TO183:LINE(0,Y)-(15,Y):NEXT:FOR Y=184TO189:LINE(0,Y)-(5,Y)
680 LINE(10,Y)-(15,Y):NEXT:RETURN
690 X=1:A$="POR ALBERTO RIVERA":FOR Y=30 TO 174 STEP 8:PSET(22,Y),4:PRINT#1,MID$
(A$,X,1):BEEP:X=X+1:FORJ=1TO30:NEXT J,Y:RETURN
700 '## CAIDA FICHA ##
710 FOR Y=1 TO 6
720 IF PO(CO,Y)<>0 THEN Y=Y-1:IF Y=0 THEN 420 ELSE 810
730 PUT SPRITE1,(CO*30+4,Y*30-46),C,1
740 PUT SPRITE0,(0,209)
750 FORI=0TO20:NEXTI
760 PUT SPRITE1,(CO*30+4,Y*30-28),C,0
770 PUT SPRITE0,(CO*30+11,-1),1,2
780 SOUND0,Y*16:FORI=0TO20:NEXTI
790 NEXT Y
800 IF Y=7 THEN Y=6
810 T=T+1:PO (CO,Y)=J
820 GOSUB 910:'SONIDO CHOQUE
830 GOSUB 1040:'PINTA LUGAR FICHA
840 GOSUB 1090:'REGISTRA 4 EN LINEA
850 IF T=54 THEN 1310 ELSE RETURN
860 '## ACORDE ##
870 SOUND 2,1:SOUND 3,3:SOUND 4,1:SOUND 5,6
880 SOUND 0,1:SOUND 1,1:SOUND 8,7:SOUND 9,7:SOUND 10,7
890 RETURN
900 '## SONIDO CHOQUE ##
910 SOUND 9,0:SOUND 10,0:SOUND 7,&B10111110:SOUND 8,31:SOUND 12,5:SOUND 13,1:RET
URN
920 '## CAMBIA COLUMNA ##
930 LINE(46,182)-(119,190),1,BF
940 PSET(50,183),1:COLORC:PRINT#1,"JUGADOR";J
950 LINE (167,182)-(248,190),1,BF
960 PSET(210-(LEN(A$(J))/2*8),183),1:PRINT#1,A$(J)
970 IF STRIG(A(J))=-1 THEN RETURN
980 A=STICK(A(J)):PUTSPRITE0,(CO*30+11,-1),1,2:IF A=0 THEN 970
990 SOUND 0,100:SOUND 1,1:SOUND 7,&B10111110:SOUND 8,31:SOUND 12,5:SOUND 13,9
1000 IF A=5 THEN RETURN
1010 IF A=3 AND CO<7 THEN CO=CO+1
1020 IF A=7 AND CO>1 THEN CO=CO-1
1030 FOR I=1TO60:NEXTI:GOTO 970
1040 '## RELLENAR AGUJERO ##
1050 CIRCLE (CO*30+15,Y*30-15),12,C
1060 PAINT (CO*30+15,Y*30-15),C
1070 PUT SPRITE 1,(0,209)
1080 RETURN
1090 '## REGISTRA LINEAS ##
1100 FOR X=1 TO 4:IFPO(X,Y)=JANDPO(X+1,Y)=JANDPO(X+2,Y)=JANDPO(X+3,Y)=JTHENX1=X+
1:Y1=Y:X2=X+2:Y2=Y:X3=X+3:Y3=Y:GOTO 1190
1110 NEXT
1120 FOR Y=1 TO 3:IFPO(CO,Y)=JANDPO(CO,Y+1)=JANDPO(CO,Y+2)=JANDPO(CO,Y+3)=JTHEN
X=CO:X1=CO:Y1=Y+1:X2=CO:Y2=Y+2:X3=CO:Y3=Y+3:GOTO 1190
1130 NEXT
1140 FOR Y=1 TO3:FORX=4TO7
1150 IFPO(X,Y)=JANDPO(X-1,Y+1)=JANDPO(X-2,Y+2)=JANDPO(X-3,Y+3)=JTHENX1=X-1:Y1=Y+
1:X2=X-2:Y2=Y+2:X3=X-3:Y3=Y+3:GOTO 1190
1160 NEXT:FOR X=1 TO4
1170 IFPO(X,Y)=JANDPO(X+1,Y+1)=JANDPO(X+2,Y+2)=JANDPO(X+3,Y+3)=JTHEN X1=X+1:Y1=Y
+1:X2=X+2:Y2=Y+2:X3=X+3:Y3=Y+3:GOTO1190
1180 NEXT:NEXT:RETURN
1190 '## GANADOR ##
1200 ON INTERVAL=30 GOSUB 1240:INTERVALON
1210 LINE (30,182)-(256,190),4,BF:LINE (62,182)-(200,190),1,BF:COLOR15:PSET (65,
183):PRINT#1,"GANO EL JUGADOR";J
1220 PLAY "T200L6V12","T200L2V9":PLAY"R806GAB07DCCED","O4GO5GE":PLAY"DGF#GDO6BGA
B","O4BO5E04E":PLAY"O7CDEDCO6BGAB","O4ABO5C":PLAY"F#GADF#AO7CO6BA","O5DF#D":PLAY
"BGABBO7DCCED","BGC":PLAY"DGF#GDO6BGAB","O4BO5ED":PLAY"EO7DCO6BAGDGF#G2","CC#DG"
1230 FOR I=1TO200:NEXTI:INTERVALOFF:GOTO 1300
1240 PUT SPRITE0,(X*30+4,Y*30-28),1,0
1250 PUT SPRITE1,(X1*30+4,Y1*30-28),1,0
1260 PUT SPRITE2,(X2*30+4,Y2*30-28),1,0
1270 PUT SPRITE3,(X3*30+4,Y3*30-28),1,0
1280 INTERVALON:FORJ=1TO40:NEXTJ
1290 PUT SPRITE0,(0,208):RETURN
1300 IF INKEY$="" THEN 1300
1310 COLOR 15,4,4:FORX=45TO233STEP30:FORY=15TO175STEP30:LINE(X-13,Y-13)-(X+13,Y+
13),4,BF:CIRCLE(X,Y),12:NEXT Y,X:LINE (49,182)-(230,190),4,BF:FORI=1TO3:PUT SPRI
TE I,(0,208):NEXTI:CLEAR:GOTO 50
1320 '## FORMA SPRITES ##
1330 DATA 15,63,127,127,255,255,255,255,127,63,31,0,0,0,0,0,192,224,224,240,
240,240,240,224,224,192,128,0,0,0,0
1340 DATA 14,63,127,127,127,31,0,0,0,14,63,127,127,63,15,0,0,128,192,224,192,128
,0,0,0,0,128,192,192,128,0,0
1350 DATA 112,112,248,112,32,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0
```

ASSEMBLER BYTES Y OTRAS YERBAS

(PARTE III)

Continuamos con la serie de artículos para poder programar en este potente lenguaje.

El set completo de las instrucciones de carga o copia de registros puede verse en la figura 1(los números a la derecha son los códigos que interpreta el microprocesador y que, en definitiva, pondremos). Podemos observar que tenemos variedad de instrucciones para manejar,y si nos detenemos en los números veremos que hay cierta relación con los registros, por ejemplo, en IX se suele usar como primer byte DDh.

Para ejecutar saltos a otras rutinas o ir a subrutinas disponemos de instrucciones encabezadas por JP (Jump),JR (Jump Relative) y CALL (llamar); el primero es equivalente al GOTO del BASIC con la salvedad de que salta a direcciones de memoria y no a líneas, el segundo también es un salto pero de espacios: es decir,si estamos en la dirección D0000 y queremos avanzar hasta la dirección D00Ah podremos hacer JP D00AH o JR0Ah indistintamente (en este último caso avanzamos diez espacios). ¿Ventajas? Vemos que el salto relativo (JR) es independiente de las direcciones de memoria y consume menos bytes de instrucción; la desventaja es que solo podemos hacerlo dentro de un margen de + - 127 bytes, por consiguiente solo sirve para saltos pequeños .

CALL es equivalente a GOSUB y respeta las mismas leyes que el BASIC; es decir, luego de ejecutar la subrutina retorna a su lugar de llamada. El set de instrucciones para estas tres instrucciones es el de la figura 2.

C = arrastre, M = signo negativo, NC = no arrastre, Z = cero, NZ = no cero, P = positivo, PO = paridad impar, PE = paridad impar .

FIGURA 1

LD (dd),A	- 32h dd	LD (dd),BC	- EDh 43h dd	LD (dd),DE	- EDh 53h dd
LD (dd),HL	- 22h dd	LD (dd),IX	- DDh 22h dd	LD (dd),IY	- FDh 22h dd
LD (dd),SP	- EDh 73h dd	LD (BC),A	- 02h	LD (DE),A	- 12h
LD (HL),A	- 77h	LD (HL),B	- 70h	LD (HL),C	- 71h
LD (HL),D	- 72h	LD (HL),E	- 73h	LD (HL),H	- 74h
LD (HL),L	- 75h	LD (IX+d),A	- DDh 77h d	LD (IX+d),B	- DDh 70h d
LD (IX+d),C	- DDh 71h d	LD (IX+d),D	- DDh 72h d	LD (IX+d),E	- DDh 73h d
LD (IX+d),H	- DDh 74h d	LD (IX+d),L	- DDh 75h d	LD A,(dd)	- 3Ah dd
LD A,(BC)	- 0Ah	LD A,(DE)	- 1Ah	LD A,(HL)	- 7Eh
LD A,(IX+d)	- DDh 7Eh d	LD A,(IY+d)	- FDh 7Eh d	LD A,A	- 7Fh
LD A,B	- 78h	LD A,C	- 79h	LD A,D	- 7Ah
LD A,E	- 7Bh	LD A,H	- 7Ch	LD A,L	- 7Dh
LD B,(HL)	- 46h	LD B,(IX+d)	- DDh 46h d	LD B,(IY+d)	- DDh 46h d
LD B,A	- 47h	LD B,B	- 40h	LD B,C	- 41h
LD B,D	- 42h	LD B,E	- 43h	LD B,H	- 44h
LD B,L	- 45h	LD BC,(dd)	- EDh 4Bh dd	LD BC,dd	- 01h dd
LD C,(HL)	- 4Eh	LD C,(IX+d)	- DDh 4Eh d	LD C,(IY+d)	- FDh 4Eh d
LD C,A	- 4Fh	LD C,B	- 48h	LD C,C	- 49h
LD C,D	- 4Ah	LD C,E	- 4Bh	LD C,H	- 4Ch
LD C,L	- 4Dh	LD D,(HL)	- 56h	LD D,A	- 57h
LD D,(IX+d)	- DDh 56h d	LD D,(IY+d)	- FDh 56h d	LD D,B	- 50h
LD D,C	- 51h	LD D,D	- 52h	LD E,E	- 53h
LD D,H	- 54h	LD D,L	- 55h	LD DE,(dd)	- EDh 5Bh dd
LD DE,dd	- 11h	LD E,(HL)	- 5Eh	LD E,(IX+d)	- DDh 5Eh d
LD E,(IY+d)	- FDh 5Eh d	LD E,A	- 5Fh	LD E,B	- 58h
LD E,C	- 59h	LD E,D	- 5Ah	LD E,E	- 5Bh
LD E,H	- 5Ch	LD E,L	- 5Dh	LD H,(HL)	- 66h
LD H,(IX+d)	- DDh 66h d	LD H,(IY+d)	- FDh 66h d	LD H,A	- 67h
LD H,B	- 60h	LD H,C	- 61h	LD H,D	- 62h
LD H,E	- 63h	LD H,H	- 64h	LD H,L	- 65h
LD HL,(dd)	- EDh 6Bh dd	LD HL,dd	- 21h dd	LD IX,(dd)	- DDh 2Ah dd
LD IX,dd	- DDh 21h dd	LD IY,(dd)	- FDh 2Ah dd	LD IY,dd	- FDh 21h dd
LD L,A	- 6Fh	LD L,B	- 68h	LD L,C	- 69h
LD L,D	- 6Ah	LD L,E	- 6Bh	LD L,(HL)	- 6Eh
LD L,(IX+d)	- DDh 6Eh d	LD L,(IY+d)	- FDh 6Eh	LD L,H	- 6Ch
LD L,L	- 6Dh	LD SP,(dd)	- EDh 7Bh dd	LD SP,dd	- 31h dd
LD SP,HL	- F9h	LD SP,IX	- DDh F9h	LD SP,IY	- FDh F9h
LD (HL),d	- 36h d	LD (IX+d),d	- DDh 36h d	LD A,d	- 3Eh d
LD B,d	- 06h d	LD C,d	- 0Eh d	LD D,d	- 16h d
LD E,d	- 1Eh d	LD H,d	- 26h d	LD L,d	- 2Eh d

FIGURA 2

JP (HL)	- E9h	JP (IX)	- DDh E9h	JP (IY)	- FDh E9h
JP C,dd	- DAh dd	JP M,dd	- FAh dd	JP NC,dd	- D2h dd
JP dd	- C3h	JP NZ,dd	- C2h dd	JP P,dd	- F2h dd
JP PE,dd	- EAh dd	JP PO,dd	- E2h dd	JP Z,dd	- CAh dd
JR C,n	- 38h n	JR n	- 18h n	JR NC,n	- 30h n
JR NZ,n	- 20h n	JR Z,n	- 28h n		
CALL C,dd	- DCh dd	CALL M,dd	- FCh dd	CALL NC,dd	- D4h dd
CALL dd	- CDh dd	CALL NZ,dd	- C4h dd	CALL P,dd	- F4h dd
CALL PE	- ECh dd	CALL PO,dd	- E4h dd	CALL Z,dd	- CCh dd
RET	- C9	RET C	- D8	RET M	- D0
RET NC	- D0	RET NZ	- C0	RET P	- F0
RET PE	- E8	RET PO	- E0	RET Z	- C8

FIGURA 3

ADC A ,(HL)	- 8Eh	ADC A,(IX+n)	- DDh 8Eh n	ADC A,(IY+n)	- FDh 8Eh n
ADC A,A	- 8Fh	ADC A,B	- 88h	ADC A,C	- 89h
ADC A,D	- 8Ah	ADC A,E	- 8Bh	ADC A,H	- 8Ch
ADC A,L	- 8Dh	ADC A,n	- CEh n	ADD A,(HL)	- 86h
ADD A,(IX+n)	-DDh 86h n	ADD A,(IY+n)	- FDh 86h n	ADD A,A	- 87h
ADD A,B	- 80h	ADD A,C	- 81h	ADD A,D	- 82h
ADD A,E	- 83h	ADD A,H	- 84h	ADD A,L	- 85h
ADD A,n	- C6h n	AND (HL)	- A6h	AND (IX+n)	- DDh A6h
AND (IY+n)	- FDh A6h	AND A	- A7h	AND B	- A0h
AND C	- A1h	AND D	- A2h	AND E	- A3h
AND H	- A4h	AND L	- A5h	AND n	- E6h n
CP (HL)	- BEh	CP (IX+n).	- DDh BEh n	CP (IY+n)	- FDh BEh n
CP A	- BFh	CP B	- B8h	CP C	- B9h
CP D	- BAh	CP E	- BBh	CP .H	- BCh
CP L	- BDh	CP n	- FEh n	DEC (HL)	- 35h
DEC (IX+n)	- DDh 35h n	DEC (IY+n)	- FDh 35h n	DEC DEC A	- 3Dh
DEC B	- 05h	DEC C	- 0Dh	DEC D	- 15h
DEC E	- 1Dh	DEC H	- 25h	DEC L	- 2Dh
INC (HL)	- 34h	INC (IX+n)	- DDh 34h n	INC (IY+n)	- FDh 34h n
INC A	- 3Ch	INC B	- 04h	INC C	- 0Ch
INC D	- 14h	INC E	- 1Ch	INC H	- 24h
INC L	- 2Ch	OR (HL)	- B6h	OR (IX+n)	- DDh 86h n
OR (IY+n)	- FDh 86h n	OR A	- B7h	OR B	- B0h
OR C	- B1h	OR D	- B2h	OR E	- B3h
OR H	- B4h	OR L	- B5h	OR n	- B5h n
SBC A,(HL)	- 9Eh	SBC A,(IX+n)	- DDh 9Eh n	SBC (IY+n)	- FDh 9Eh n
SBC A,A	- 9Fh	SBC A,B	- 98h	SBC A,C	- 99h
SBC A,D	- 9Ah	SBC A,E	- 9Bh	SBC A,H	- 9Ch
SBC A,L	- 9Dh	SBC A,n	- DEh n	XOR (HL)	- AEh
XOR (IX+n)	- DDh AEh n	XOR (IY+n)	- FDh AEh n	XOR A	- AFh

Ahora analizaremos el grupo aritmético y lógico que se divide en dos grupos: de 8 y 16 bits.

De 8 tenemos: Suma, suma con arrastre, resta, resta con arrastre, comparaciones (siempre compara con el contenido del acumulador), incrementos (le suma uno al registro que corresponda) y las operaciones lógicas and, or, exor (respecto del acumulador). Son las de la figura ADC = suma con arrastre, ADD = suma, INC = incrementa en uno, CP = compara, DEC = decrece en uno, SBC = resta con arrastre.

En 16 bits es más pobre la lista ya que solo podemos sumar, restar,decrecer e incrementar. (Ver figura 4). Pero además de operaciones matemáticas, el hecho de operar directamente en Assembler nos permite manejar los bytes con más facilidad que con cualquier lenguaje de alto nivel, por ejemplo correr un bit a la izquierda de la palabra (byte) o rotarlo a la derecha, (figura 5). El set completo es el de la figura 6.

FIGURA 4

ADC HL,BC	- EDh 4Ah	ADD HL,BC	- 09h	ADD IX,BC	- DDh 09h
ADC HL,DE	- EDh 5Ah	ADD HL,DE	- 19h	ADD IX,DE	- DDh 19h
ADC HL,HL	- EDh 6Ah	ADD HL,HL	- 29h	ADD IX,IX	- DDh 29h
ADC HL,SP	- EDh 7Ah	ADD HL,SP	- 39h	ADD IX,SP	- DDh 39h
ADD IY,BC	- FDh 09h	ADD IY,DE	- FDh 19h	ADD IY,IY	- FDh 29h
ADD IY,SP	- FDh 39h	DEC BC	- 0Bh	DEC DE	- 1Bh
DEC HL	- 2B	DEC IX	- DDh 2Bh	DEC IY	- FDh 2Bh
DEC SP	- 3Bh	INC BC	- 03h	INC DE	- 13h
INC HL	- 23h	INC IX	- DDh 23h	INC IY	- FDh 23h
INC SP	- 33h	SBC HL,BC	- EDh 42h	SBC HL,DE	- EDh 52h
SBC HL,HL	- EDh 62h	SBC HL.SP	- EDh 72h		

FIGURA 5

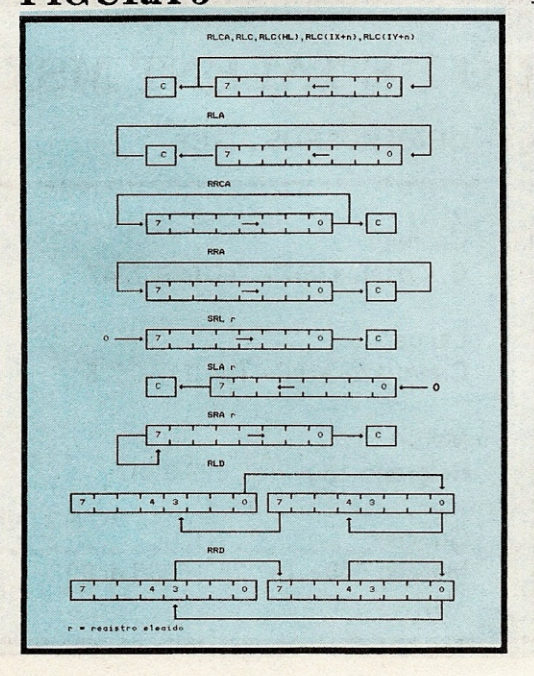


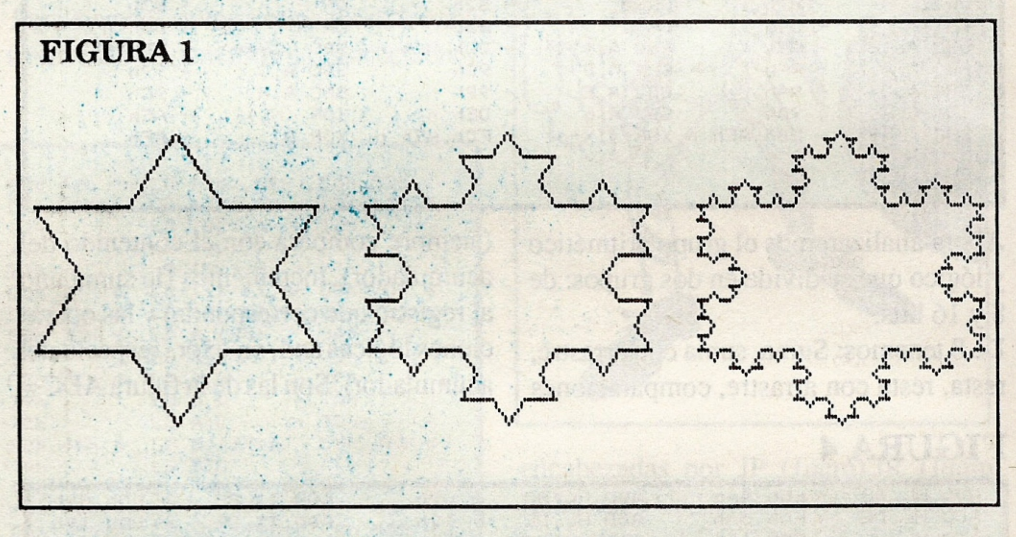
FIGURA 6

RL (HL)	- CBh 16h	RL(IX+n)	-DDh CBh n 16h	RL(IY+n)	-FDh CBh n 16h
RL A	- CBh 17h	RL B	- CBh 10h	RL C	- CBh 11h
RL D	- CBh 12h	RL E	- CBh 13h	RL H	- CBh 14h
RL L	- CBh 15h	RLA	- 17h	RLC (HL)	- CBh 06h
RLC(IX+n)	-DDh CBh n 06h	RLC(IY+n)	-FDh CBh n 06h	RLC A	- CBh 07h
RLC B	- CBh 00h	RLC C	- CBh 01h	RLC D	- CBh 02h
RLC E	- CBh 03h	RLC H	- CBh 04h	RLC L	- CBh 05h
RLCA	- 07h	RLD	- EDh 6Fh	RR (HL)	- CBh 1Eh
RR(IX+n)	-DDh CBh n 1Eh	RR(IY+n)	-FDh CBh n 1Eh	RRA	- CBh 1Fh
RR B	- CBh 18h	RR C	- CBh 19h	RR D	- CBh 1Ah
RR E	- CBh 1Bh	RR H	- CBh 1Ch	RR L	- CBh 1Dh
RRA	- 1F	RRC (HL)	- CBh 0Eh	RRC(IX+n)	-DDh CBh n 0Eh
RRC A	- CBh 0Fh	RRC B	- CBh 08h	RRC(IY+n)	-FDh CBh n 0Eh
RRC C	- CBh 09h	RRC D	- CBh 0Ah	RRC E	- CBh 0Bh
RRC H	- CBh 0Ch	RRC L	- CBh 0Dh	RRC A	- 0Fh
RRD	- EDh 67h	SLA (HL)	- CBh 26h	SLA(IX+n)	-DDh CBh n FEh
SLA A	- CBh 27h	SLA B	- CBh 20h	SLA(IY+n)	-FDh CBh n FEh
SLA C	- CBh 21h	SLA D	- CBh 22h	SLA E	- CBh 23h
SLA H	- CBh 24h	SLA L	- CBh 25h	SRA (HL)	- CBh 2Eh
SRA A	- CBh 2Fh	SRA B	- CBh 28h	SRA(IX+n)	-DDh CBh n 2Eh
SRA C	- CBh 29h	SRA D	- CBh 2Ah	SRA(IY+n)	-FDh CBh n 2Eh
SRA E	- CBh 2Bh	SRA H	- CBh 2Ch	SRA L	- CBh 2Dh
SRL (HL)	- CBh 3Eh	SRL A	- CBh 3Fh	SRL(IX+n)	-DDh CBh n 3Eh
SRL B	- CBh 38h	SRL C	- CBh 39h	SRL(IY+n)	-FDh CBh n 3Eh
SRL D	- CBh 3Ah	SRL E	- CBh 3Bh	SRL H	- CBh 3Ch
SRL L	- CBh 3Dh				

FRACTALES EN LOGO

***Lucas Film Inc., el estudio cinematográfico que realiza los efectos especiales de las películas de George Lucas, utiliza para ello una de las más modernas computadoras.
Una de las técnicas para generar gráficos usando computadoras es con fractales. De esta manera es posible la generación de montañas, lagos, nubes, y otros muchos elementos de la escenografía.***

Benoit Mandelbrot, matemático francés, ha pasado gran parte de su vida estudiando las complejas formas de la naturaleza y buscando un procedimiento para reproducirlas. En este camino, Mandelbrot creó un nuevo objeto geométrico: los FRACTALES (del latín, irregular o fragmentado). El fractal se suma a los objetos de la Geometría Euclidiana: el punto, la recta, el plano, etc.
Un fractal nunca puede ser dibujado, sólo puede ser representado aproximadamente.



LA CURVA COPO DE NIEVE.

Para comprender mejor el procedimiento, dibujémosla primero con lápiz y papel.

Paso 1: Dibujemos un triángulo equilátero.
Paso 2: Dividamos cada uno de sus lados en 3 partes iguales.
Paso 3: Marquemos el segmento central de cada lado.
Paso 4: Construyamos con los segmentos del paso 3, triángulos equiláteros orientados hacia afuera.
Paso 5: Borremos los segmentos marcados en el paso 3 .
Paso 6: Volvamos al paso 2 .
En la Figura 1 vemos las tres primeras figuras generadas por este procedimento.

CENTROS DE ASISTENCIA AL USUARIO DE TALENT MSX

CAPITAL FEDERAL

Centro Cultural de la Ciudad de Buenos Aires
Taller Logo de computación: Junín 1930

Fundación de Informática y Educación
Centro de Computación Clínica Asistencia al Usuario Discapacitado: Ramsay 2250 - Pabellón F
Tel. 784-2018

Barrio Norte
Uriburu 1063 - Tel. 83-6892/826-6692

Belgrano
Mendoza 2728 - Tel. 781-2271

Centro
Av. Córdoba 654 - Tel. 392-5328/7611/8043/8051/8251

Flores
Gral. Artigas 354 - Tel. 612-3902

Palermo
Guatemala 4733 - Tel. 71-4124

San Telmo
Chile 1345 - Tel. 37-0051 al 54

GRAN BUENOS AIRES

Castelar
C. Casares 997 - Tel. 629-2247

Lanús
Caaguazú 2186 - Tel. 247-0678

Morón
Belgrano 160 - Tel. 629-3347

Quilmes
Moreno 609 - Tel. 253-6086 al 89

Es fácil ver que éste no termina nunca. Por eso es que un fractal sólo puede ser aproximado, nunca dibujado completamente.

Dibujemos ahora con LOGO una aproximación del copo de nieve:

```
para copo.1 :t :d
iz 30
repetir 6 [copo.2 :t :d]
de 30
fin
para copo.2 :t :d
si igual? :d 1 [ad :t iz 60 ad
:t de 120 parar]
copo.2 :t / 3 :d - 1
copo.2 :t / 3 :d - 1
de 180
copo.2 :t / 3 :d - 1
copo.2 :t / 3 :d - 1
fin
para copo
bp
sp
at 60
cp copo.1 36 3
fin
```

Ahora escribamos copo y veremos el último dibujo de la Figura 1.

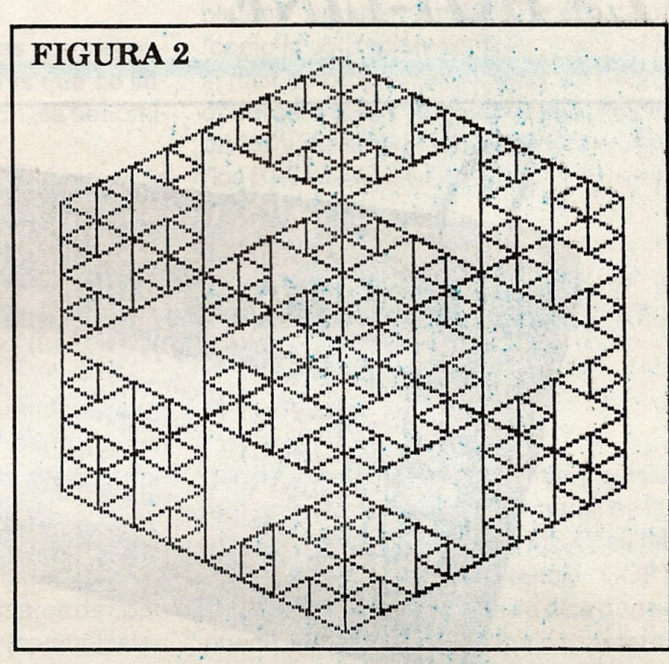
Podemos experimentar dándole otros valores a copo.1 en copo y ver qué sucede. El primer número indica el tamaño, y el segundo el equivalente al número de repeticiones del procedimento realizado con lápiz y papel.

Para dibujar el fractal completo este segundo número tendría que ser infinito.

Veamos otro ejemplo con LOGO:

```
para tri
bp
activar [0 1 2 3 4 5]
mt
cada [de quien * 60]
tri.1 60
ot
fin
para tri.1 :l
si :l < 4 [parar]
repetir 3 [tri.1 :l / 2 ad :l.de
120]
fin
```

FIGURA 2



Ahora escribamos

tri

Y veremos en la pantalla la Figura 2.

Un último ejemplo:

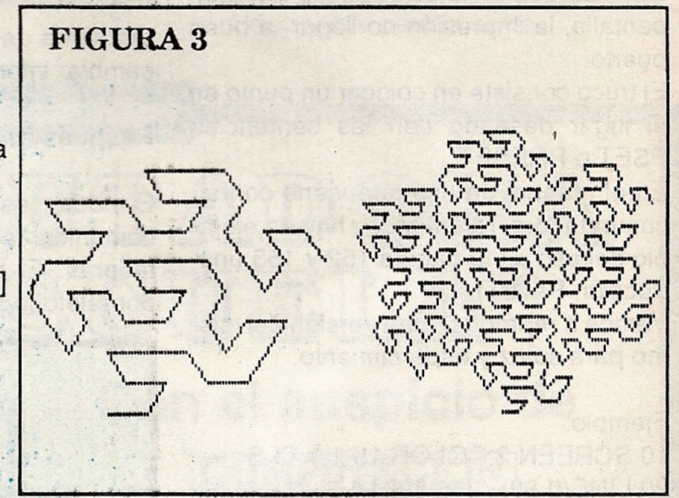
```
para gosper :t :d :i
si :d = 1 [gosper.l :t :i parar]
gosper :t :d - 1 1 iz 60
gosper :t :d - 1 -1 iz 60 * :i
gosper :t :d - 1 -1 de 60 * :i
gosper :t :d - 1 :i de 60 * :i
gosper :t :d - 1 1 iz 60 * :i
gosper :t :d - 1 1 de 60
gosper :t :d - 1 -1
fin
para gosper.l :t :i
ad :t iz 60
si :i = 1 [ad :t] [ad :t * 2]
iz 120 ad :t de 60 * :i ad :t de 120
si :i = 1 [ad :t * 2] [ad :t]
de 60 ad :t
fin
```

Escribamos ahora:

gosper 5 3 1

y en nuestra pantalla aparecerá el último dibujo de la Figura 3.

FIGURA 3



BIBLIOGRAFIA

-JUEGOS DE ORDENADOR
A.K. DEWDNEY. I
INVESTIGACION Y CIENCIA
Octubre de 1985

-CREATING FRACTALS
W. Mc WORTER/J. TAZELAAR
BYTE
Agosto 1987

-MATEMATICA E IMAGINACION (II)
E. KASNER/J. NEWMAN

CONTESTANDO LA HOT-LINE

Pregunta 1:

(Cómo logro imprimir un texto en las pantallas gráficas SCREEN 2 y SCREEN 3? He intentado hacerlo con la sentencia PRINT sin ningún resultado.

Respuesta:

El problema está en que la sentencia PRINTsolo es válida en las pantallas de texto (SCREEN 0 y 1). Para poder operar en los modos gráficos se debe abrir un archivo denominado "GRP:" y manejar directamente la impresión como se hace con cualquier archivo.
La impresión se efectuará a partir del último punto mostrado en la pantalla. Esto significa que si no se imprimió nada en pantalla, la impresión no llegar a buen puerto.
El truco consiste en colocar un punto en el lugar deseado con las sentencias PSET o PRESET.
En el manual de uso que viene con su computadora Talent MSX, hay un ejemplo del uso en la página 152 y 153 (instrucción PAINT).
Vamos a reproducir una versión del mismo para solaz y esparcimiento.

Ejemplo:

```
10 SCREEN 2:COLOR 15,1,1:CLS
20 LINE(0,88)-(255,88),14
30 LINE(129,88)-(18,192),14
40 LINE(129,88)-(200,192),14
50 PAINT(110,110),14
60 LINE(0,89)-(255,89),12
70 LINE(121,89)-(5,192),12
80 LINE(137,89)-(215,192),12
90PAINT(10,120),12:PAINT(210,120),12
100 LINE(0,87)-(255,87),7
110FORI=1TO70:PSET(INT(RND
(1)*255),INT(RND(1)*80)),15:NEXT
 120 OPEN "GRP:" FOR OUTPUT AS ¿1
130 FOR I=0 TO 80 STEP 2
140 C=C+1:IF C<3 OR C>15 THEN C=3
150 PSET(I*1.5+50,(9-I/4)^2),1:COLOR
C:PRINT ¿1,"TALENT MSX"
160 NEXT I:COLOR 15,4,4
170 GOTO 170
```

Pregunta 2:

¿Si conecto a mi Talent MSX con otras computadoras vía modem, tengo algún problema de compatibilidad para intercambiar información?



Respuesta:

El modem es un dispositivo que permite comunicar telefónicamente dos computadoras. El proceso de comunicación por teléfono se puede representar así:

1. El computador que emite, utiliza un modem para enviar los datos. Los datos se envían en forma de tonos sonoros a través del teléfono. Este proceso se denomina "modulación".
2. ENTel transmite los tonos (Más o menos fielmente...)
3. El computador que recibe, utiliza un modem para retraducir los tonos a seña-

le s que la computadora "entiende". Este proceso se denomina "demodulación".
Viendo el proceso de "MOdulación/DE-Modulación" vemos de dónde surge el nombre del modem...
Los modem utilizan distintas normas para comunicarse. Las normas que se impusieron a nivel mundial son las denominadas "CCITT" y "Bell".
La norma que más difusión tiene es la CCITT (que es la aceptada oficialmente por ENTel).
También es importante la velocidad de transferencia, que normalmente varía entre 300 y 1200 baudios (baudio =aprox 1 bit p/segundo).
Además existen otros parámetros que no detallaremos aquí, pero que hacen que la comunicación llegue a feliz término.
Una vez puestos de acuerdo en todo lo anterior, ya se puede comunicar con su amigo, aunque tenga otra marca de computadora. Puede dialogar (comunicarse como si fuera un telex), enviarse mutuamente archivos (en formato ASCII) e intercambiar programas (siempre que sean computadoras de la misma marca o norma).
Por ejemplo, si comunicamos una Talent MSX con una TI-99/4A no tendrán problemas en intercambiar textos o datos (como la cotización del dolar, p.ej.), pero si reciben programas escritos en BASIC deberán trabajar bastante para modificarlos y hacerlos "ejecutables" en MSX (los BASIC de ambas máquinas son diferentes).
Por último, si se desea intercambiar programas escritos en código de máquina, ambas computadoras deben ser del mismo tipo.

Pregunta 3:

Tengo una Talent MSX2 Turbo y quisiera saber lo siguiente: en el manual de las extensiones al BASIC para MSX2 se hace referencia a la sentencia COPY SCREEN. En ese manual se dice que se puede tomar una fuente de video externa para digitalizar imágenes. ¿Puedo utilizar una video-cassettera para ingresar la imagen? ¿Cómo la conecto?

Respuesta:

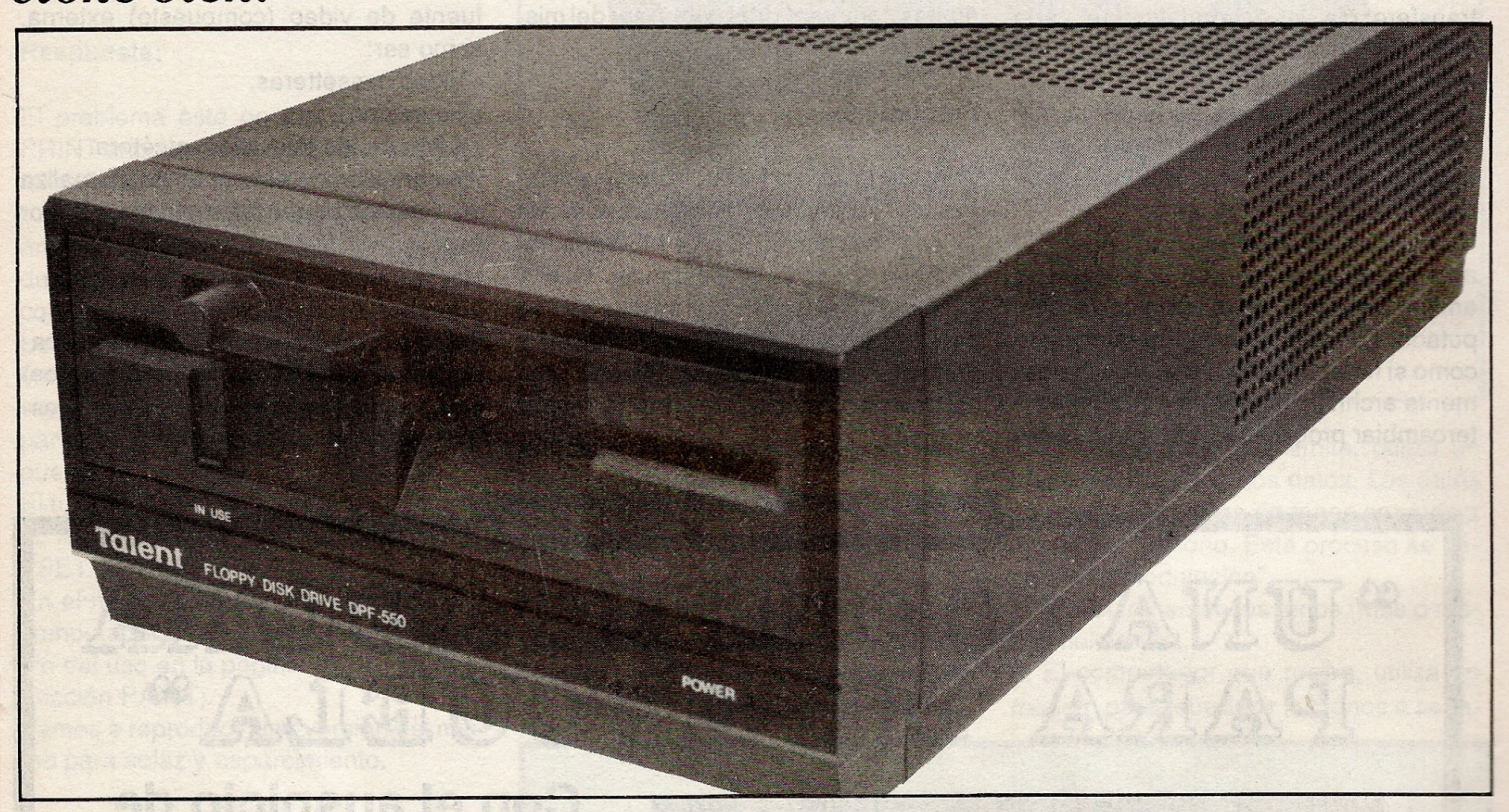
Parafraseando al manual, vemos que COPY SCREEN sirve para "digitalizar una fuente de video externa, **usando un equipo periférico adecuado.**
El equipo periférico a que hace referencia el manual se denomina "Interfaz digitalizadora y superimposición de imágenes". Dicho periférico permite conectar su Talent MSX2 Turbo con cualquier fuente de video (compuesto) externa, como ser:
- Video cassetteras.
- Televisión.
- Cámaras de televisión, etcétera.

La conexión de este perifrico se realiza vía el conector que denominamos "Color bus".
Hasta ahora, todo muy lindo, pero... ¿dónde se consigue?
Este periférico esta en la fase de desarrollo en Talent Computación (nosotros) y se estima que promediando el año estará disponible para la venta.

LOS MISTERIOS DEL MSX-DOS: OPERACION

(primera parte)

Uno de los primeros problemas con que tropieza el usuario de MSX surge cuando decide comprar una unidad de disquetes. Y no es que dicha unidad presente habitualmente fallas o no funcione bien.



Ocurre que el entorno de trabajo del MSX-DOS es bastante diferente del de casete, y los términos nuevos que debe manejar el sufrido usuario son algo que debe incorporar paulatinamente, a fin de poder utilizar correctamente su nuevo equipamiento.

Si hemos comprado una unidad de discos Talent MSX DPF-550, la primer recomendacón quedebemos respetar es:

"Lea cuidadosamente el Manual de Uso"

El mismo ha sido escrito íntegramente en el país (o sea, está en castellano) y hemos tratado de incorporar todos los "chimentos útiles" para el máximo aprovechamiento del equipo.

En esta nota continuaremos con el análisis del otro lado del sistema, desde el punto de vista del usuario avanzado que compró su unidad de discos, para que quede claro el entorno de trabajo que brinda el MSX-DOS y de esta forma aprovechar sus excelentes recursos.

COMANDOS EXTERNOS

Creando un comando externo

Antes de comenzar a bucear en el formato de grabación del MSX-DOS, veamos cómo crear rápidamente un comando externo.

El MSX-DOS maneja, vía el COMMAND.COM, dos tipos de comando:

* Comandos internos
* Comandos externos

Los comandos internos vienen incorporados en el COMMAND.COM y se ejecutan directamente. Por ejemplo, DIR o COPY.

Los comandos externos existen en el disquete como archivos con la extensión ".COM" e ingresando el nombre del comando externo (sin la extensión) hace que el comando se ejecute de la siguiente forma:

1. Se carga el comando externo desde 100h en adelante.
2. Se llama a 100h

Los programas desarrollados en lenguaje ensamblador que trabajen en la direcciónde memoria 100h y almacenados con nombres de archivo con la

FIGURA 1

```
LD  E,0Ch     ;E:= código de control del CLS
LD  C,02h     ;C:= nro. de función para CONSOLE OUTPUT (salida por consola)
CALL 0005h    ;call al BDOS
RET
```

FIGURA 2

```
10 '* Este programa crea "CLS.COM"
20
30 OPEN "CLS.COM" FOR OUTPUT AS #1
40
50 FOR I=1 TO 8
60     READ D$
70     PRINT #1,CHR$(VAL("&H"+D$));
80 NEXT
90
100 DATA 1E,0C,0E,02,CD,05,00,C9
```

extensión ".COM" se denominan comandos externos y se pueden ejecutar desde MSX-DOS.

Por ejemplo, consideremos un programa que envía el carácter de control "0Ch" a la pantalla mediante la rutina de salida de carácter y borra la pantalla. Este es el programa de 8 bytes correspondiente la figura 1.

Escribiendo estos 8 bytes en un archivo llamado CLS.COM hace que el comando externo "CLS" borre la pantalla, cuando se lo soliciten. El siguiente programa utiliza el acceso a archivos secuenciales del Disk-BASIC para crear el comando. Luego de ejecutar es de programa, el comando CLS queda almacendo en el disquete. Confirme que la rutina funciona regresando al MSX-DOS (con CALL SYSTEM)Ver figura 2.

Pasando argumentos a un comando externo.

Cuando creamos un comando externo, existen dos formas de pasar argumentos desde la línea de comando al comando externo. Primero, cuando pasa los nombres de archivos en la línea de comandos como argumentos, utiliza las posiciones 5Ch y 6Ch como área de trabajo del sistema.

El COMMAND.COM, que siempre considera al primer y segundo parámetro como nombres de archivos cuando se ejecutan comandos externos, los expande hasta completarlos incluyendo el número de drive (1 byte) + nombre del archivo (8 bytes) + extensión (3 bytes) y los almacena en 5Ch y 6Ch. Este formato es el correspondiente a los primeros 12 bytes del FCB (File Control Block, Bloque de Control de Archivo), lo que hace más fáciles ciertas operaciones para utilizar el FCB. Más adelante veremos el concepto de FCB.

Sin embargo, este método utiliza solamente 16 bytes de diferencia entre los dosFCBs; ya sea 5Ch o 6Ch, deben utilizarse en forma exclusiva como un FCB

TABLA 1

```
*-----------------*----------------------------------------------------*
| Boot sector     | Programa de carga del MSX-DOS e informaci"n        |
|                 | propia del disco.                                  |
*-----------------*----------------------------------------------------*
| FAT             | Control f/sico de la informaci"n sobre los         |
|                 | datos en el disco.                                 |
*-----------------*----------------------------------------------------*
| Directorio      | Informaci"n de control de los archivos en el       |
                                 26
                  | disco.                                             |
*-----------------*----------------------------------------------------*
| Area de datos   | Datos de los archivos propiamente dichos.          |
|
*-----------------*----------------------------------------------------*
```

FIGURA 3

```
            /*------------------* <- sector # 0
            | | boot sector      |
            | *------------------* <- sector # ?\
     Area   | |      FAT         |                |
    total   | *------------------* <- sector # ? > Los sectores tope
      del   < |   directorio     |                | de esta área se pueden
            | *------------------* <- sector # ?/   obtener con el DPB
    disco   | |     Area de      |
            | |      Datos       |
            | |                  |
            \*------------------* <- #ltimo sector
```

completo Por lo tanto, el COMMAND.COM además de utilizar las direcciones 5Ch y 6Ch, también almacena la l!nea de comando completa, excluyendo el propio nombre del comando externo, en el formato:

cantidad de bytes (1 byte) + línea de comando y de esta forma, el comando externo puede interpretar correctamente los argumentos.

Esta información se almacena en el área del DMA (normalmente 80h) aunque se puede cambiar esta dirección. También dejamos para un futuro no muy lejano la explicación del DMA.

ESTRUCTURA DE LOS ARCHIVOS DE DISCO.

La información sobre la estructura de los datos en el disco y cómo se los contro la es importante cuando se accede al disco utilizando las "llamadas al sistema".

Comenzaremos con una descripción de los "sectores lógicos" que son las unidades mínimas de intercambio de datos entre el disco y el MSX-DOS, y luego continuaremos con el manejo de datos dentro de "archivos", que es algo más familiar para los programadores.

Unidades de datos en el disco

***Pistas:** Los discos musicales utilizan pistas para almacenar el registro musical. Existe un error común al considerar que son muchas las pistas (o surcos) que contiene un disco cuando en realidad es una sola pista: comienza en el borde del disco y se va reduciendo hasta el final del mismo, formando un espiral.

En cambio, en un disquete las pistas no son espiraladas sino concéntricas. A cada pista le corresponde una posición del motor paso-a-paso que controla el cabezal de lecto-grabación de la unidad de disco.

Cada uno de estos anillos concéntricos se denomina "pista". La unidad de discos Talent DPF-550/555 utiliza 40 pistas por lado.

*** Sectores:** MSX-DOS puede acceder a muchos tipos de unidades de discos incluyendo a los disquetes de 3.5 " 2DD y los discos duros. Para poder manejar los distintos tipos de medios de almacenamiento, las "llamadas al sistema" consideran los "sectores lógicos" como la unidad mínima de datos en el disco. Un sector lógico se especifica con un número que comienza en el 0.

El sector lógico es una subdivisión de la pista. Cada pista en la DPF-555 contiene 9 sectores.

*** Clusters:** Mientras se utilicen las "llamadas al sistema", un sector puede ser la unidad mínima de datos, como se describió anteriormente. Sin embargo, la realidad es que los datos son manejados en unidades de "clusters", compuestos de varios sectores. Como describiremos en la sección destinada al FAT, un cluster se especifica con un número que comienza en 2 hasta el tope del área de datos que se especifica en el cluster número 2.

Nuevamente, en la DPF-550/555 se utilizan clusters que contienen 2 sectores.

Haciendo némeros vemos que si cada sector contiene 512 bytes, un cluster contiene 1024 bytes (1 kbyte) y el disquete formateado contiene 360 kbytes.

*** Conversión de clusters a sectores:** En una parte del directorio o el FCB, que describiremos luego, vemos que la ubicación de los datos se indica con los clusters.

Para utilizar las llamadas al sistema indi-

FIGURA 4

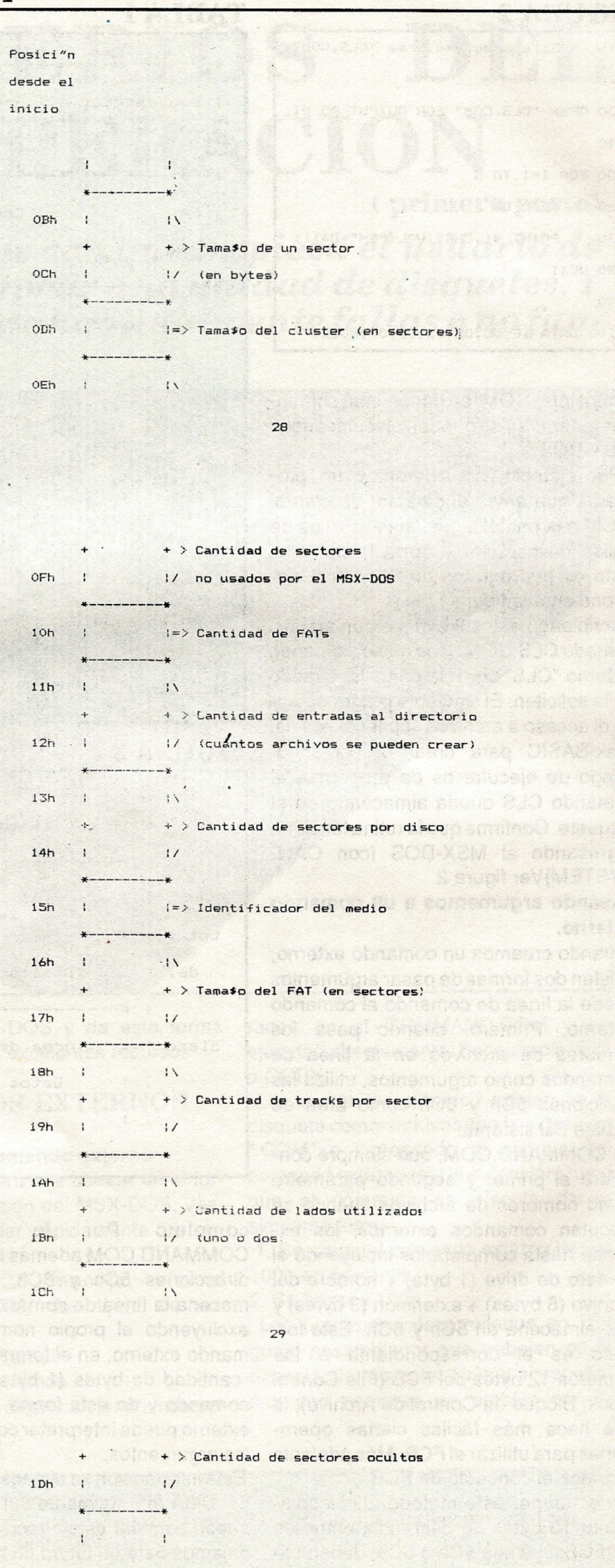


FIGURA 5

```
          *----------*
BASE=>    |          |-> N#mero de drive
          *----------*
+1        |          |-> Identificador del medio
          *----------*
+2        |          |\
          +          + > Tama$o del sector
+3        |          |/
          *----------*
+4        |          |-> Máscara del directorio
          *----------*
+5        |          |-> Desplazamiento del directorio
          *----------*
+6        |          |-> Máscara de cluster
          *----------*
+7        |          |-> Desplazamiento de cluster
          *----------*
+8        |          |\
          +          + > Sector tope de FAT
+9        |          |/

                    30

          *----------*
+10       |          |-> Cantidad de FATs
          *----------*
+11       |          |-> Cantidad de entradas de directorio
          *----------*
+12       |          |\
          +          + > Sector tope del área de datos
+13       |          |/
          *----------*
+14       |          |\
          +          + > Cantidad de clusters + 1
+15       |          |/
          *----------*
+16       |          |-> Cantidad de sectores requeridos por un FAT
          *----------*
+17       |          |\
          +          + > Sector tope del área de directorio
+18       |          |/
          *----------*
+19       |          |\
          +          + > Direcci"n en memoria del FAT
+20       |          |/
          *----------*
```

cadas con cluster, la relación existen te entre el cluster y el sector debe ser calculada.

Dado que el cluster Nro. 2 y el sector máximo de datos se ubican en la misma posición, se puede calcular el número de sector partiendo del número de cluster de la siguiente forma:

1. Supongamos que el cluster que tenemos sea el número C.
2. Examinemos el sector tope del área de datos (leyendo el DPB - Drive Parameter Block (Bloque de parámetros de la unidad de discos-) y asumamos que es S0.
3. Examinemos el número de sectores equivalentes a un cluster (usando la función1Bh) y asumamos que es n. En el caso del DPF-555 es n=2.
4. Utilice la fórmula:

S=S0+(C-2)*n para calcular el número de sector.

En el MSX-DOS, los sectores del disco se dividen en 4 áreas, como vemos en la Tabla 1. El cuerpo de datos de un archivo se escribe en el disco en la porción den ominada "área de datos".

La información necesaria para poder manipular estos datos se escribe en el disco en tres áreas. La figura 3 nos muestra la relación de las ubicaciones de estas áreas.

El boot sector (sector de arranque) se ubica siempre en el sector 0, pero los se ctores topes (FAT, directorio y área de datos) difieren según sea el medio; por lo tanto, se debe utilizar el DPB para saber su ubicación.

*** DPB (drive parámeter block - bloque de parámetros de la unidad de discos) y boot sector:**

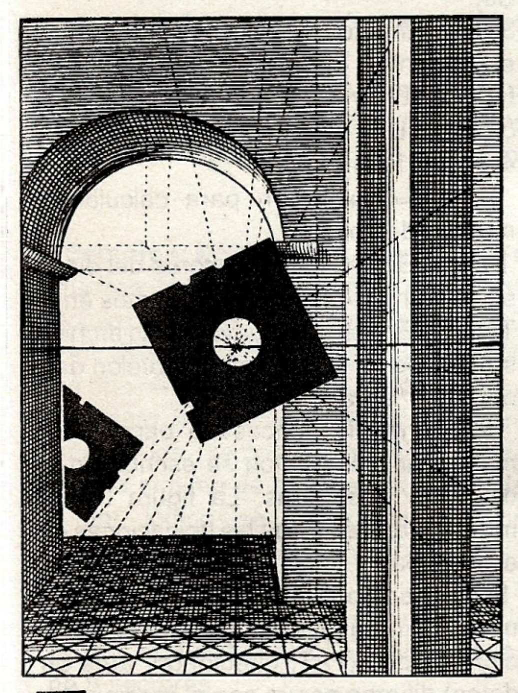
En el MSX-DOS, el área "DPB" se almacena en el área de trabajo en memoria de cada unidad de discos conectada, y la información propia de cada unidad se almacena allí. MSX-DOS puede manejar la mayoría de los tipos de discos, porque las diferenciasentre los medios de almacenamiento se zanja con el uso del DPB, que permite que el MSX-DOS maneje las posiciones relativas de los distintos componentes descriptivos del disco.

La información escrita en el DPB, que originalmente se encuentra en el boot sector (sector de inicio, que es el sector 0) se lee al inicializarse el MSX-DOS. Véase la diferencia entre los contenidos del boot sector y el DPB en las figuras 4 y 5.

HUGO DANIEL CARO

HARDCOPY *CLASE: UTILITARIO*

Este programa se encarga de copiar una pantalla gráfica al papel con una impresora matricial.



El funcionamiento del programa se basa en la lectura de la memoria de pantalla carácter por carácter y, por medio de una transformación conveniente, de enviarlo a la impresora común.

La computadora arma la imagen en la pantalla según el ordenamiento de la figura 1;cada bloque se compone de 64 celdillas en grupos de 8,dispuestos horizontalmente (figura 2).Cada celdilla (pixel) al iluminarse genera una imagen que en conjunto forma el dibujo.El método que usa para formarla es el siguiente: la arma carácter por carácter empezando por la esquina superior izquierda y hacia la derecha;cada uno se compone a su vez de 8 bloques de 8 pixels que van ubicados de arriba hacia abajo,luego sigue con el caracter de al lado, etcétera (figuras 2 y 3).

La impresora,en cambio,toma el bloque de forma vertical y va armando el carácter horizontalmente de izquierda a derecha hasta completar la línea (ver figura 4).

Lo que el programa hace es mandar los comandos para convertir la impresora a modo gráfico y luego transforma el for-

```
                  ; HCOPY1 V 1.3
                  ;
                  ; 26 / 05 / 87
                  ;
                  ;
                  ;
                  ; NOTA:
                  ;       Este programa esta preparado
                  ;       para funcionar en una impre-
                  ;       sora SEIKOSHA GP-550A o simi-
                  ;       lar.
                  ;
                  ;
                  ; USO:
                  ;       Se usa en modo 2 de Screen;se agrega
                  ;       al programa DEFUSR=40000:? USR (X),
                  ;       donde X es un parámetro de escala y
                  ;       y vale de 1 a 4 ( según el ancho ).
                  ;       toma por defecto el valor 2.
                  ;
                  ;
                  ;
                  ;
                  ;
                  ;
                  ;RUTINAS DE BIOS
                  ;
                  ;
                  ;
                  ;LPTOUT:
                  ;       Imprime un caracter en
                  ;       printer dato en Acc.
                  ;
                  ;LDIRMV:
                  ;       Copia de RAM de video
                  ;       en RAM de basic bytes.
                  ;       HL=Direccion de VRAM
                  ;       DE=Direccion de RAM
                  ;       BC=Cantidad de bytes
                  ;
                  ;
                  ;
00A5 =            LPTOUT  EQU     00A5H
                  ;
0059 =            LDIRMV  EQU     0059H.
                  ;
                  ;
                  ;
                  ORG 40000
                  ;
                  ;
                  ;
                  ;
                  ;
9C40 3E00                 LD A,0            ;
9C42 013000               LD BC,30H         ;
9C45 2AF8F7               LD HL,(0F7F8H)    ;Lectura
9C48 BD                   CP L              ;de la
9C49 C24F9C               JP NZ,JUMP3       ;escala
9C4C 210200               LD HL,0002H       ;si no
9C4F 09           JUMP3:  ADD HL,BC         ;asume 2
9C50 22DB9C               LD (WRAM),HL      ;
9C53 210000               LD HL,0           ;Contador (A) de
9C56 3E18         LOOP4:  LD A,24           ;0 a 6143 en pasos
9C58 BC                   CP H              ;de 256 bytes
9C59 C8                   RET Z             ;Si HL=6144 RETURN.
9C5A 22DF9C               LD (VRAM),HL      ;Lo guarda en VRAM.
9C5D 3E1B                 LD A,1BH          ;       Adapta el
9C5F CDA500               CALL LPTOUT       ;       "LINE FEED"
9C62 3E39                 LD A,39H          ;       a modo
9C64 CDA500               CALL LPTOUT       ;       Grafico.
9C67 110000               LD DE,0           ;       Contador (B) de
9C6A 3EF0         LOOP3:  LD A,240          ;       0 a 256 en pasos
9C6C BB                   CP E              ;       de 8 bytes. Si DE
9C6D CAC79C               JP Z,JUMP1        ;        =256 nueva linea.
9C70 ED53DD9C             LD (CRAM),DE      ;       Lo guarda.
9C74 2ADF9C               LD HL,(VRAM)      ;            (1) Compone la
9C77 19                   ADD HL,DE         ;            direccion del
9C78 11E19C               LD DE,TRAM        ;            caracter de VRAM
9C7B 010800               LD BC,8           ;            y lo trae a RAM
9C7E CD5900               CALL LDIRMVH      ;            de trabajo(TRAM),
9C81 110000               LD DE,0           ;            Contador (C) de 0 a 7
9C84 3E08         LOOP2:  LD A,8            ;            en paso de 1 byte.
9C86 BB                   CP E              ;            Si E=8 nuevo
9C87 CABA9C               JP Z,JUMP2        ;            caracter.
9C8A 21E19C               LD HL,TRAM        ;                 Trasformada de
9C8D 0608                 LD B,8            ;                 la matriz del
9C8F 4E           LOOP1:  LD C,(HL)         ;                 caracter de
9C90 CB11                 RL C              ;                 8 x 8 pixels.
```

```
9C92 1F                  RRA             ;           Provoca la
9C93 71                  LD (HL),C       ;           impresion a lo
9C94 2C                  INC L           ;           ancho de la
9C95 10F8                DJNZ LOOP1      ;           hoja.
9C97 47                  LD B,A          ;           Prepara a la
9C98 3E1B                LD A,1BH        ;           impresora

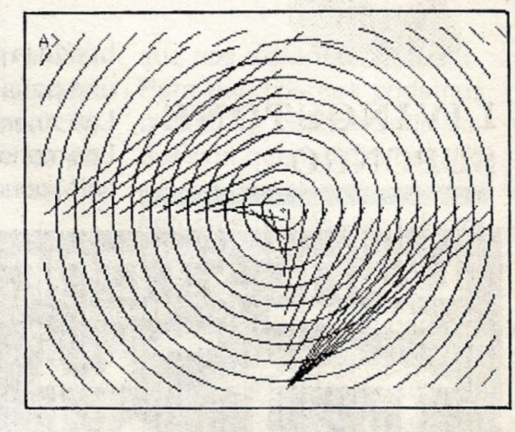

9C9A CDA500              CALL LPTOUT     ;           para dos
9C9D 3E56                LD A,56H        ;           funciones:
9C9F CDA500              CALL LPTOUT     ;           La pone en
9CA2 3E30                LD A,30H        ;           modo grafico
9CA4 CDA500              CALL LPTOUT     ;           y lee un
9CA7 3E30                LD A,30H        ;           parametro de
9CA9 CDA500              CALL LPTOUT     ;           repeticion
9CAC 3ADB9C              LD A,(WRAM)     ;           (escala) que
9CAF CDA500              CALL LPTOUT     ;           le pasamos
9CB2 78                  LD A,B          ;           desde el basic.
9CB3 CDA500              CALL LPTOUT     ;           Imprime caracter
9CB6 13                  INC DE          ;       Incrementa (C) y
9CB7 C3849C              JP LOOP2        ;       vuelve.
9CBA ED5BDD9C   JUMP2:   LD DE,(CRAM)    ;    Incrementa (B)
9CBE EB                  EX DE,HL        ;    vuelve.
9CBF 010800              LD BC,8         ;
9CC2 09                  ADD HL,BC       ;
9CC3 EB                  EX DE,HL        ;
9CC4 C36A9C              JP LOOP3        ;
9CC7 3E0D       JUMP1:   LD A,0DH        ;    Manda a impresora
9CC9 CDA500              CALL LPTOUT     ;    un "PRINTING".
9CCC 3E0A                LD A,0AH        ;    Manda a impresora
9CCE CDA500              CALL LPTOUT     ;    un "LINE FEED".
9CD1 2ADF9C              LD HL,(VRAM)    ;Recupera valor
9CD4 010001              LD BC,100H      ;en VRAM incre-
9CD7 09                  ADD HL,BC       ;menta (A) en
9CD8 C3569C              JP LOOP4        ;256 y vuelve.
                ;
                ;
                ;
                ;
                ;
                ;ZONA DE TRABAJO
                ;
                ;
                ;
                ;
9CDB            WRAM:    DEFS    2       ;Memoria de parametro
9CDD            CRAM:    DEFS    2       ;Memoria de vaiable contador
9CDF            VRAM:    DEFS    2       ;Guarda la direccion de VRAM en curso
9CE1            TRAM:    DEFS    8       ;Zona de deposito del caracter llamado
```

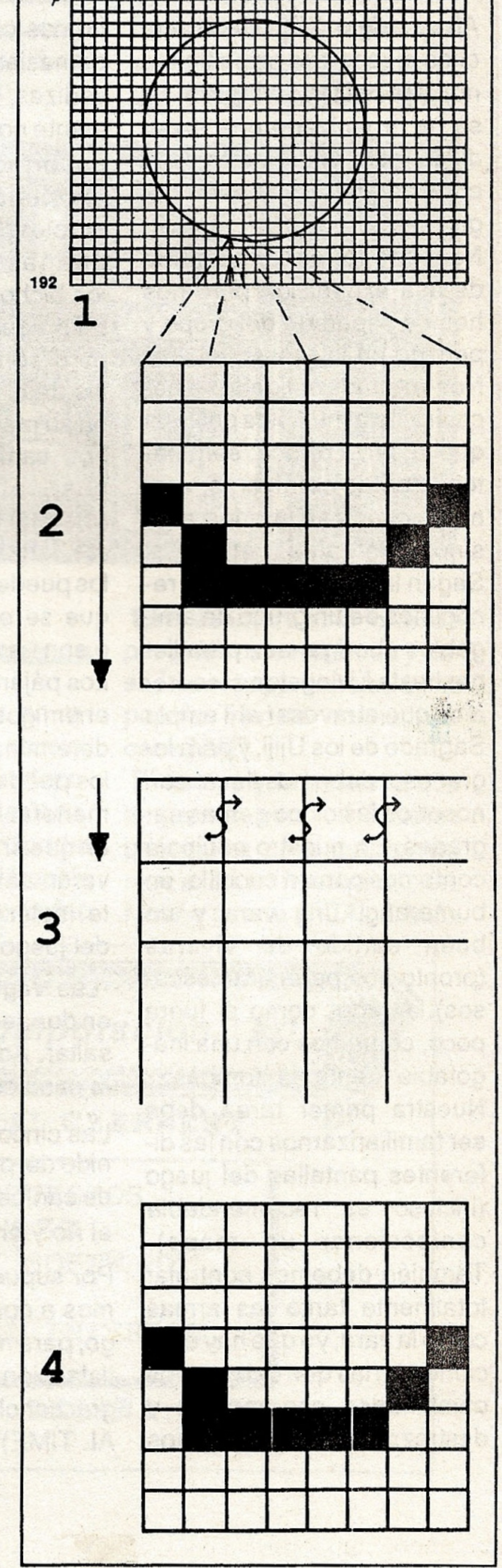
mato del carácter para que luego pueda ser interpretado por la impresora y así dibujado.

Se muestra aquí el listado fuente en mnemónico para ser ensamblado con GENS, DUAD, ZEN, o cualquier otro ensamblador. Si no se dispone de uno hay un programa para cargarlo en la dirección 40000 y luego guardarlo como SAVE "HCOPYI.OBJ", 40000, 40300.

Se deben llamar como LOAD "HCOPYI.OBJ" y luego de haber hecho o cargado el dibujo (previo SCREEN 2). Como ejemplo aquí hay un programa que dibuja unas circunferencias y unas líneas formando efectos dimensionales, hay que observar que la rutina en cuestión está a partir de la línea 60000, es decir que se puede agregar a cualquier programa como una rutina auxiliar.

```
10 SCREEN 2
20 CLEAR200,39999!
30 FOR A=1 TO 180 STEP 10
40 CIRCLE (128,96),A
50 NEXT A
60 FOR A=1 TO 255 STEP 10
70 LINE(A,96)-(128,A)
80 NEXT A
60000 '
60001 '
60002 '    RUTINA DE HARDCOPY
60003 DEFUSR = &H9C40
60004 FOR N = &H9C40 TO &H9CDA
60005 READ D$:POKE N,VAL("&H"+D$)
60006 NEXT N
60007 PRINT USR (2)
60008 DATA 3E,0,1,30,0,2A,F8,F7,BD,C2,4F,9C,21,1,0,9,22,DB
60009 DATA 9C,21,0,0,3E,18,BC,C8,22,DF,9C,3E,1B,CD,A5,0,3E,39
60010 DATA CD,A5,0,11,0,0,3E,F0,BB,CA,C7,9C,ED,53,DD,9C,2A,DF
60011 DATA 9C,19,11,E1,9C,1,8,0,CD,59,0,11,0,0,3E,8,BB,CA
60012 DATA BA,9C,21,E1,9C,6,8,4E,CB,11,1F,71,2C,10,F8,47,3E,1B
60013 DATA CD,A5,0,3E,56,CD,A5,0,3E,30,CD,A5,0,3E,30,CD,A5,0
60014 DATA 3A,DB,9C,CD,A5,0,78,CD,A5,0,13,C3,84,9C,ED,5B,DD,9C
60015 DATA EB,1,8,0,9,EB,C3,6A,9C,3E,D,CD,A5,0,3E,A,CD,A5
60016 DATA 0,2A,DF,9C,1,0,1,9,C3,56,9C
```





LIVINGSTONE, SUPONGO



Africa, siglo XIX. Hace muchos meses que no se tienen noticias del Dr. Livingstone y se ha organizado una expedición para encontrarlo en el corazón del continente negro.

Nosotros formábamos parte de esa expedición pero nos hemos separado del grupo y perdido en la selva.

Nos veremos obligados a hacer frente a múltiples peligros que la selva ofrece: serpientes, plantas carnívoras, arenas movedizas, monos asesinos,caníbales, etcétera. Según la información que recogimos de un grupo de amigables aborígenes , para llegar hasta Livingstone tendremos que atravesar el Templo Sagrado de los Ujiji, y para lograr eso, debemos llevar con nosotros las cinco gemas sagradas. En nuestro equipaje contamos con un cuchillo, un bumerang, una vara y un buen surtido de víveres (pronto nos parecerán escasos). Aparte, como si fuera poco, contamos con una inagotable fuente de bombas.

Nuestra primer tarea debe ser familiarizarnos con las diferentes pantallas del juego (incluso es recomendable confeccionar un mapa). También debemos controlar totalmente tanto las armas como la vara, ya que hay ocaciones en las que deberemos combinarlas con rapidez y destreza. La tecla "H" nos brinda la posibilidad de hacer una pausa en el juego

Los enemigos son muchos. Los monos, asesinos por instinto, ponen continuamente a prueba su puntería a base de "cocazos" sobre nuestra humanidad. Las plantas carnívoras son indestructibles, por lo que no nos conviene acercarnos demasiado. Las arenas movedizas, en las cuales obviamente no debemos caer. Los escorpiones, murciélagos, serpientes, y pirañas son los bichos que más abundan, pero son fáciles de eliminar en su mayoría. Los caníbales nos atacan constantemente con hechizos, lanzas y cerbatanas. Se los puede matar. Las sirenas, que se encuentran en el río,son amores que matan. Los pájaros, nuestros peores enemigos. Sólo aparecen en determinadas pantallas y se los puede matar de cualquier manera. El problema radica en que si nos agarran nos llevarán a su nido, cosa bastante molesta a ciertas alturas del juego.

Las vagonetas de la mina, en donde el único remedio es saltar. Además los gases de la mina son muy peligrosos.

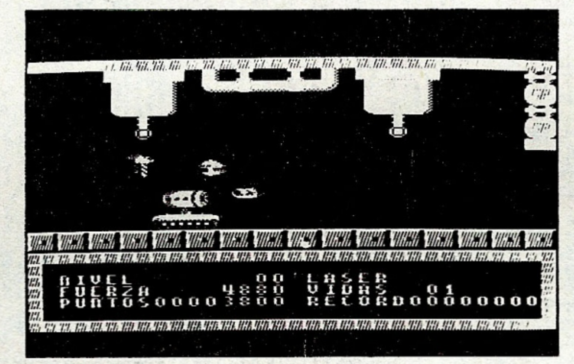
Las cinco gemas están en el nido del pájaro, en el poblado de caníbales, en la cueva, en el río y en la mina.

Por supuesto que no les vamos a contar el final del juego, para no privarlos de la satisfacción del premio por lograr concluir la aventura (REAL TIME)

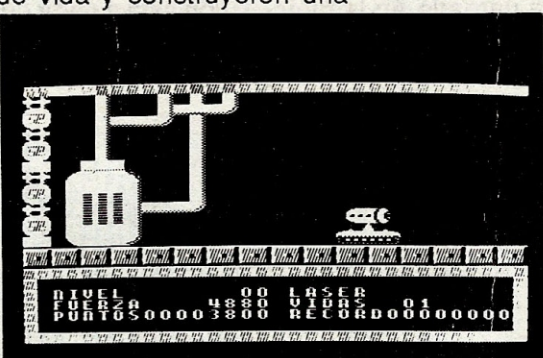
THE LAST MISSION

Hace siglos, muchos siglos, la Tierra fue víctima de su propia automatización. El ser humano no cejó en su empeño de crear robots cada vez más inteligentes y poderosos, hasta que llegó el día en que su grado de perfección fue tal que el hombre fue presa de su propio progreso.

Las máquinas más inteligentes iniciaron lo que en su época se denominó "la revolución de las máquinas".



Levantaron contra la humanidad a los restantes autómatas, consiguieron apoderarse de la Tierra, terminar con la supremacía del hombre y destruir la naturaleza. Hicieron desaparecer de la faz del planeta todo vestigio de vida y construyeron una gigantesca central, base energética de su poderío.



La única esperanza de los supervivientes humanos, que consiguieron asentarse en el planeta Nova, está en su última creación: El robot "OR-CABE-3". Su misión en la Tierra comienza en el interior de la Gran Central, de la que ha robado los planos de defensa. Debe subir nivel por nivel, luchando contra sus enemigos, hasta salir a la superficie .Luego deberá llegar hasta la nave que lo retornará al planeta Nova , para que desde allí el hombre, con los planos secretos, pueda destruir la tiranía de la inteligencia artificial que sojuzga la Tierra.

El "OR-CABE-3" consta de dos partes diferentes: una oruga locomotora y base de su energía, y una cabeza propulsora que se acopla a la oruga, poseedora de un mortífero láser para destruir a los enemigos. (REAL TIME)

HISTORIAS CLINICAS

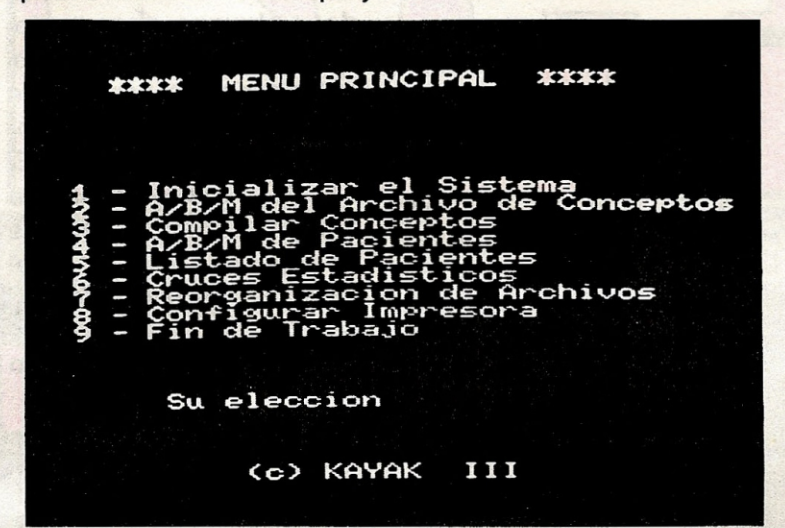
Este sistema permite desarrollar la historia clínica de todos y cada uno de los pacientes de un hospital, sanatorio o clínica privada de acuerdo a los valores de seguimiento que establezca el usuario y obtener todo tipo de información, inclusive la del tipo estadístico.

La característica del sistema es que es el mismo profesional el que define qué datos va a utilizar y cómo.

El sistema viene con archivos de demostración que permiten al usuario practicar a gusto, sin temor a los errores que pueda cometer.

La información de cada paciente puede dividirse en dos tipos de datos. Por un lado, los datos que son fijos (nombre, fecha de la próxima cita, etcétera), o que pese a variar

en el tiempo no interesa mantener todos los valores sino solamente el actual (dirección, antecedentes familiares, etcétera). Por otro lado, aquella información que puede variar en el tiempo y



para la cual es necesario guardar todos los valores que va asumiendo. Por tal motivo existen datos que son únicos para cada paciente y otros que se almacenan para cada consulta, entrevista o estudio.

El menú general presenta las opciones principales del sistema: altas, bajas y modificaciones del archivo de items, en el que el profesional define cuáles son los datos que necesita para una explotación óptima del sistema.

Compilar conceptos: este programa toma los datos del archivo de items para generar los archivos donde estarán todos los datos de los pacientes.

Altas, bajas y modificaciones de pacientes, Listados de pacientes son otras de las funciones del sistema.

Cruces estadísticos: se pueden realizar por pacientes y por casos. Se entiende por cruce por paciente cuando se cumple una condición una o más veces en un paciente, pero se considera como una sola. En cruces por casos se considera cada ocurrencia de la condición aunque sea dentro del mismo paciente.

El sistema procesará todos los datos de todos los pacientes que posea en ese momento y listará los resultados obtenidos: cantidad de pacientes tratados, cantidad de casos tratados (si se tomó la opción por casos), cantidad de veces que se cumplieron las condiciones requeridas y, en caso de haberlo solicitado, número de todos los pacientes que cumplieron las condiciones.

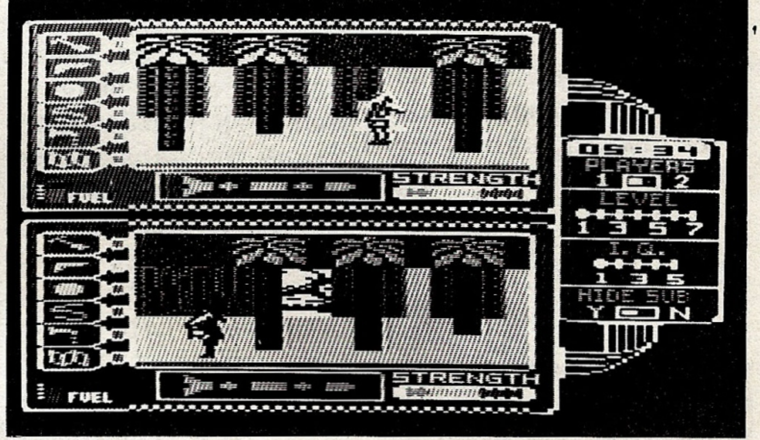
Finalmente, el sistema posee una opción para reorganizar los archivos existentes, eliminando todos los registros que corresponden a números de historias clínicas anuladas.

Creemos que este utilitario será de un valor destacado para los profesionales del área médica. (KAYAK)

que nos espera en la costa. Para defendernos podemos utilizar los objetos sofisticados con que cuenta nuestro equipo. Por ejemplo, una poderosa arma láser que lanza efectivas granadas (de vez en cuando).

La vida de nuestro personaje está representada en la pantalla por el marcador de fuerza; cuando llegue a "D", nuestra aventura habrá finalizado.

Podemos reponer energía, evitando las trampas y localizando las piezas del misil.



SPY VS SPY

La aventura consiste en recuperar las piezas de un misil que los malvados han depositado en una isla desierta. Con el misil en nuestro poder tenemos que ingeniárnosla para encontrar el submarino

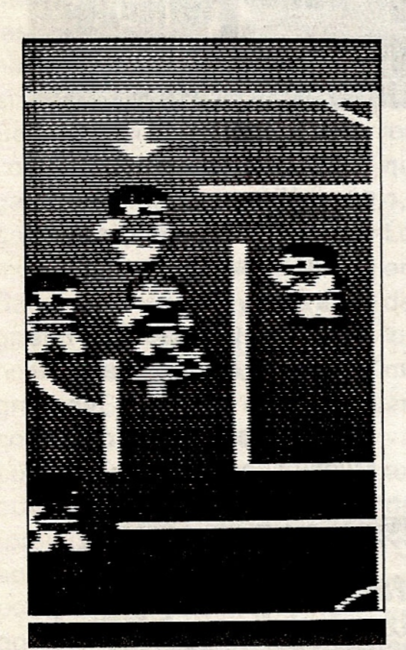
Esta aventura está formada por gráficos excelentes y basada en los personajes de la popular revista norteamericana MAD.

El tiempo y la destreza que tengamos son los factores principales para que tengamos éxito en nuestra misión. Mucha suerte (REAL TIME).

DUDAS GRAFICAS

1.-¿Es posible realizar efectos de scrolling de pantalla utilizando sentencias como LINE, DRAW, etc. ?
2.-¿Se pueden realizar SPRITES en forma multicolor sin definir diferentes SPRITES de un color y que forman parte de un dibujo ?
3.-¿Existe alguna manera de lograr que cuando se realizan SPRITES formados por más de una retícula de 16x16 pixels, no se muevan a destiempo y parezca que la figura esta formada por partes ?

MARCOS R. OGGERO
RAFAELA - STA. FE



Load MSX

Primeramente debemos aclarar que las sentencias que nos mencionas no corresponden a manejos de pantalla relacionados con movimientos de la misma, como se entiende bajo la palabra scrolling. Dicha instrucción se encuentra incluida en el lenguaje incorporado de otras marcas y se utiliza como auxiliar de presentacion de impresiones en pantalla; con ella se consigue un "efecto de levantamiento de persiana" de los textos. Pese a todo, no debe lamentarse la falta de dicha instrucción, ya que puede simularse mediante rutinas escritas en lenguaje ensamblador y, por otra parte, el sistema operativo se encarga de efectuar el scroll cuando es necesario.Respecto de los SPRITES, debe consignarse que no es posible definir SPRITES multicolores en la MSX I, pero sí es posible hacerlo en la MSX versión II; por lo tanto la única manera de realizarlos es la mencionada en tu carta. Igualmente, los SPRITES sufren un efecto de desplazamiento debido a los tiempos de ejecución que provocan dichos retrasos en la impresión de cada uno de ellos ; una mejora al problema sería la utilización de rutinas escritas en lenguaje de máquina.

Para comunicarse con nosotros deben escribirnos a "Load MSX", Paraná 720 ,5º Piso, (1017), Capital Federal.

DRIVES INTERCAMBIABLES

Poseo una computadora TOSHIBA HX-20 MSX y quisiera que me aclararan algunos puntos que paso a detallar :
1.-Por lo que pude averiguar hasta el momento en las casas del ramo , no existen en el mercado periféricos de la marca TOSHIBA y además los representantes de la marca TOSHIBA han bajado sus persianas, ya que esta empresa ha cerrado en el Japón.
2.-Si ya no hay periféricos, quisiera saber si puedo utilizar una unidad de discos TALENT tal como viene o debo agregarle algo para su uso.
3.-De no ser posible el uso de esta unidad , ¿qué otra marca puedo adquirir?.

CARLOS A. CURO
CAP. FED.

Load MSX

Actualmente casi no quedan periféricos de dicha marca en el mercado porque la empresa representante efectivamente ha cerrado sus puertas ; la segunda parte de la pregunta no podemos contestarla, ya que no tenemos confirmación al respecto. Cualquier unidad de discos es, teóricamente, compatible con cualquier consola de otra marca; pero debe hacerse una salvedad respecto a dicho punto , no todos los fabricantes respetan la norma MSX en su totalidad y esto implica la necesidad de precaución en su interconexión.Las marcas mencionadas son compatibles en forma directa, ya sea a través de un cartucho adaptador para el conector superior o en el conector de expansiones trasero.
Otras marcas deben ser confirmadas en su conexión.

CASETE MISTERIOSO

Quiero consultarles debido a algunas observaciones hechas al reproducir el sonido de los casetes sin el cable de grabador conectado en el conector EAR 1._¿Qué representan los tonos cambiantes que se escuchan ? 2.-¿El nivel de salida es muy alto? 3.-Quisiera armar un cable más largo. ¿Puedo hacerlo yo?

MARIANO ARIEL ROSA

Load MSX

Los tonos representan niveles lógicos ; es decir , traducido para la máquina, cada frecuencia corresponde a un estado en particular . Un "uno" equivale a 2400 Hz. (vibraciones por segundo) , en tanto que un "cero" lo es a un tono de 1200 Hz. ; para los más entendidos diremos que la modulación es en sistema FSK(Frecuency Shift Keying)
No , está adaptado para poder conectarse a la mayoría de los reproductores de casete ,suponemos que te refieres al nivel de salida de la máquina , éste se encuentra entre los 7 y 22 mV PAP (pico a pico)
Es posible armarlo fácilmente comprando un cable de ocho conductores ,dos fichas tipo Spica de 3,5 mm de diámetro , una de 1,5 mm y otra del tipo DIN 45326 .
Esta última tiene las patitas numeradas para facilitar las conexiones a saber : 1-2-3-8 = MASA ; 4 = SALIDA ; 5 = ENTRADA ; 6 = CONTACTO + DEL MOTOR ; 7 = CONTACTO - DEL MOTOR .





PARA TODOS

AÑO 3 Nº 35 # 12.- REP. ARGENTINA

ROBOTS
¿AMIGOS O ASESINOS?

SPECTRUM
GRAFICAR ES FACIL

DEPORTES
POR COMPUTADORA

INEDITOS
10 PROGRAMAS

NUEVO CONCURSO
Y SORTEOS

SPECTRUM - CZ - TK - DREAN COMMODORE - TI - MSX - ATARI y PC

Encienda una computadora Talent MSX y sus periféricos.

MODEM
- Interfaz asincrónica incorporada.
- Normas BELL y CCITT, 300 y 1.200 baudios.
- Incluye en 80 Kb software de comunicaciones, MSX-PLAN y MSX-WRITE.

UNIDAD DE DISCOS FLEXIBLES
- 5 1/4" de 360 Kb (DS-DD)
- Velocidad transferencia 250 Kb/sg.
- Formato grabacion compatible MS-DOS.

MINI-LAN
EQUIPO CENTRAL PROFESOR

IMPRESORA

EQUIPO ALUMNO 1 2 3 4 5 6 7 8 9 10

DISKETTERAS 2 x 360 KB

EQUIPO ALUMNO

(RED PARA USO EDUCACIONAL)
- Comunicación por línea compartida a 30.000 baudios.
- Comparte unidades de disco e impresoras de consola MSX maestro con hasta 10 consolas MSX alumno
- Carga simultánea de un programa a todos los alumnos.
- Carga individual de maestro a alumno
- Salvado de programa alumno a unidad de disco maestro.
- Salida a impresora de maestro del listado de programa alumno, en spooling.
- Funciona desde MSX-Basic, MSX-Logo o cualquier programa que corra desde Basic.
- Estando activa, se dispone de todo el MSX-Basic.

RS-232
- Velocidades programables desde 50 a 19.200 baudios.
- Procesamiento de comunicaciones desde Basic.
- Buffer de 128 caracteres para recepción

TECLADO NUMERICO
- Conexion a Joystick
- Se integra a todo el software que corre bajo MSX-DOS (Ej.: D-Base II, MSX-Plan, etc.)

MOUSE
- Código DPM-220, accesorio para graficar.

MONITOR MONOCROMATICO 12"
- Anti-reflejo - Fósforo verde.
- Apto para uso profesional.
- (80 caracteres x 25 líneas).
- Parlante con amplificador incorporado.

EXPANSION 80 COLUMNAS
- Hace posible la utilización de software estandar CP/M, emulando terminal tipo VT-52.
- Incluye software para manejo de video.

Software

MSX-LOGO	MSX-LPC	MSX-PLAN	MSX-WRITE
Desarrollado por Logo Computer System Inc. con aplicación de primitivas y redacción del Manual por los Ings. Hilario Fernández Long y Horacio Reggini.	Lenguaje de programación estructurado y en castellano.	Planilla de calculo de Microsoft Corp. (Versión para MSX del Multiplan.)	Procesador de palabra de ASCII Corp. en castellano.



Talent
Tecnología y Talento *en su casa*

Producida en San Luis por Telemática S.A. licenciataria exclusiva de Microsoft Corp. y ASCII Corp. para uso de la norma MSX en Argentina.
6 meses de garantía y mensualmente en su quiosco la revista Load MSX.